Preface

Some five years ago, this work started from the observation that a *lack of autonomy* — or *intelligence* – is a major obstacle for a break-through of mechatronic systems such as manipulators or mobile robots in the areas where their potential flexibility is of enormous industrial and social importance: tele-operated tools to replace or support the human worker in hazardous environments, semi-automatic and supervised assistants to disabled people, highly-autonomous machines in moderately structured industrial settings, etc. In these five years, the main cause of this autonomy defect has not changed: the engineering community has insufficient knowledge and expertise to integrate the *sensing*, *planning* and *control* aspects of an autonomous system. There are lots of good sensors on the market, humans excel in problem solving, and systems theory and control have proven to be *the* engineering disciplines of the twentieth century; nevertheless, no task controller software exists yet that can make a machine mimic the human capacity to interact with the world around him.

Models are believed to be in the centre of the above-mentioned integration of planning, sensing and control: they help the planner to focus its attention, they provide the deterministic setting in which control theory feels comfortable, and they put up the framework in which to transform sensor signals from raw *data* into useful *information.* This last argument is fundamental to the work described here, because it was felt as a huge disadvantage of the current state-of-the-art in compliant motion —i.e., the hybrid or impedance set-point control paradigms, and the Task Frame motion specification formalism— that they return only low-level data to the user —measured forces or motions— while even the most modest intelligent behaviour requires the ability to reason on high-level information about the contact situation —what are the current motion constraints, and where do they occur?

This thesis presents a modest contribution to bridging the enormous intelligence gap mentioned above. The modesty lies in the fact that it restricts its application area to

1. *compliant robot motion*, i.e., a force controlled robotic arm holds a tool —the so-called *manipulated object*— with which it has to perform a task that brings it in contact with its environment. (Remark, however, that this text does *not* want to propose (another) new force control algorithm!)

2. *geometric modelling* of the instantaneous motion constraints that appear on the tool. The models contain *first* and *second order* parameters, i.e., they describe which velocities (i.e., first order motion parameters) the manipulated object can execute in order to move while maintaining the contact, as well as which accelerations (second order motion parameters) are compatible with the constraint. Geometrically, this corresponds to knowing the tangent directions at the contact points (first order geometric parameters), as well as the local *curvature* (second order geometric parameters).

3. derivation of *identification algorithms.* The knowledge of, on the one hand, the instantaneous geometric parameters of a contact situation, and, on the other hand, the specified motion of the manipulated object, suffices to make a reasonable prediction of how the contact geometry, as well as the reaction forces, will evolve over time. Of course, the real contact geometry deviates from the modelled contact geometry. Hence, the predicted evolution does not accurately correspond to the measured forces and motions. The main contribution of this text is the development of equations —called *identification equations*— that link these deviations between model and measurements to estimates of the errors in the first and second order geometric parameters.

It is important to interpret correctly the meaning of the words "instantaneous" and "identification" as they are used in the previous paragraphs: this text does *not* aim at finding out which contact situation —to be chosen from a set of possible hypotheses— is currently constraining the manipulated object's motion; it does want to contribute to the problem of resolving —on-line, iteratively and for multiple contacts at the same time— the uncertainties about the exact geometry of a *given type* of contact situation. That is, the type of the contacts is known —e.g., two *vertex-surface* contacts between the manipulated object and the environment— but the position of the contact points, the direction of the contact normals, as well as the curvature parameters can be wrong. The errors in these parameters are estimated based on 1) the current geometry of the contact model, and 2) the latest motion and force measurements.

During the thesis work, we have been constantly and consistently emphasizing the following approach: 1) keep the models as *general* as possible, for a theory that applies to all possible contact situations is to be preferred over a non-coherent set of ad hoc approaches; 2) keep the models as *simple* as possible, because complex models tend not to be used in practice, even if they describe the physical world more accurately; 3) provide robust and efficient on-line sensor processing techniques to extract sufficient information from the raw sensor data for continuous adaptation of the models to the actual contact situation.

The text describes an implementation of this approach; the result is called *kinematic modelling for robot compliant motion.* Another shorthand name —the *Contact Frame Formalism*— was chosen to indicate its complementarity with respect to the state-of-the-art *Task Frame Formalism.* The main idea is to consistently replace the contacts on the manipulated object by "virtual contact manipulators," which constrain the manipulated

object in the same way as the contacts. The result is a kinematic chain, with a branch for each contact, and also one for the robot arm. This kinematic approach complies with the requirements of generality and simplicity, since kinematics is one of the most mature research areas in robotics. The second major and fully original contribution of this thesis supports the fulfilment of the third requirement: make the model on-line adaptable by providing identification equations that estimate the model errors.

Judging from the ever more abundant robotics literature, the Robotics Research group of our Department has been among the first and is still among the very few to investigate the area described above —i.e., on-line identification of geometric uncertainties for force controlled compliant motion— at least at a level beyond that of estimating the local contact normal from the measured force. I have also been lucky to start my research career in the Esprit Basic Research Action *FIRST*, Fundamentals of Intelligent Robot Systems —some years later continued in its follow-up project, co-financed by the Flemish Government, with the original name *SECOND*, Sensory Controlled Dextrous Robots— in which other research groups work in the very complementary domains of *planning* of compliant motions, and *reasoning* about contact situations. My research work profited also from the Belgian Programme on Interuniversity Attraction poles initiated by the Belgian State — Prime Minister's Office, Science Policy Programming (IUAP-50)— especially by providing people and material to help me do real world experiments.

Research is, fortunately, not performed in isolation. Hence, I enormously appreciate the social and technical inputs from the many members of the PMA family. In particular, I have spent many pleasant and interesting moments with the "control guys" Dirk Torfs and Jan Swevers and "handy-man" Dominiek Reynaerts; with Peter Van de Poel and Wim Witvrouw, who were my first thesis students, and also the ones that gave me most; with Wim Persoons, the most practical doctoral student in the lab, and in this respect a welcome antipole to my own research work; with my invaluable office mates Sabine Demey, Stefan Dutré and Marnix Nuttin —they were always ready for critical brainstorming discussions, and for trying to prove me wrong; moreover, Sabine was a tremendous support during many hours of frustrating and/or successful robot programming— and with Bagus Arthaya, the perfectly integrated embodiment of Eastern inspiration and friendliness. Of course, I have profited a lot by the experience of some senior members of our research and administrative staff, so many sincere thanks go to: the "one-and-only" Hans Thielemans, who makes life much easier by always knowing practical solutions to all practical problems; Paul and Philippe Vanherck, for being indispensable and almost feeling guilty about it; Jan Thielemans, Paul Van Cauwenbergh, Raymond Wuyts, Ann Letelier, Lieve Notré, Karin Dewit, Carine Coosemans, Luc Haine, Fernand Van Meeuwen, Paul Bielen, Dirk Bastiaensen, Frans De Kelver, Jean-Pierre Merckx and Firmin Vandevelde, for so much practical assistance and good humour; and Paul Valckenaers (founder of *World Travel, Inc.* and *Speech Unlimited*), who did a very good job in convincing me to stay at the Department.

I also have tremendous appreciation for our "founding fathers" Rik Van Brussel, and,

especially,[1] Joris De Schutter, since they offered me the most important things in a professional life: opportunities, challenges and responsibilities.

Finally —it's only a small sentence for a man, but of a giant importance for his well being— a sincere and respectful *thank you* to my parents, Greet and Ine, Marina, Erik, Jacques and Pól, for the most important things in a human life: love, happiness and understanding.

J.E.D.

H.B.
Leuven
17 Feb. 1995

[1] This is an understatement!

Abstract

This thesis is about *force controlled compliant robot motion*, with the emphasis on: 1) *modelling* of arbitrary and time-varying contact situations between a rigid manipulated object and rigid objects in its environment, 2) *motion specification* in terms of allowed velocities and accelerations for the manipulated object, maintaining the contact with the physical constraints but without generating too large contact forces, and 3) *on-line identification of uncertainties* in the instantaneous geometric parameters of the motion constraint model, i.e., the position of the contact points, the direction of the contact normal, and the local curvature parameters. Requirements for generality, simplicity and robustness have guided the research work.

The main new *theoretical results* of this thesis are: 1) the successful use of the *kinematics of mechanisms* as the unifying framework for the integration of modelling, specification and identification; 2) the use of the first and second order geometric parameters of the contacting surfaces of the manipulated object and its environment as basic model parameters; 3) the derivation of identification equations for uncertainties in these parameters, based on the general physical properties of consistency and reciprocity; and 4) new symbolical algorithms to calculate the derivative of the Jacobian matrix of a kinematic chain with respect to one of the chain's joint angles.

Real-world application of these theoretical results has been proven feasible by various experiments on a commercial robot arm equipped with force control software of moderate complexity.

Beknopte samenvatting

Deze thesis handelt over *krachtgecontroleerde robots*, die taken uitvoeren waarbij *contacten* met de omgeving optreden. De nadruk ligt op : 1) het *modelleren* van willekeurige tijdsvariante contactsituaties waarbij zowel het gemanipuleerde object als de omgeving uit starre lichamen bestaan; 2) het opstellen van een *bewegingsspecificatie* in termen van snelheden en versnellingen die aan het gemanipuleerde object kunnen opgelegd worden in de gegeven contactsituatie zonder contact te verliezen enerzijds, en anderzijds zonder te grote reactiekrachten op te wekken; en 3) het *identificeren* van *onzekerheden* in het contactmodel, in reële tijd tijdens de taakuitvoering. De vereisten voor algemene geldigheid, eenvoud

van concepten, en robuustheid tegen stoorinvloeden vormden steeds de leidraad voor het onderzoek.

De belangrijkste *theoretische* resultaten van deze thesis zijn : 1) de *kinematica van mechanismen* is op een doorgedreven wijze gebruikt als het unificerende kader waarin modelleren, specificeren en identificeren zijn verenigd; 2) de contactmodellen zijn gebaseerd op de eerste en tweede orde geometrische parameters van de oppervlakken in contact; 3) de begrippen van consistentie en reciprociteit vormen de fysische onderbouw bij het afleiden van de identificatievergelijkingen; en 4) voor de berekening van de afgeleiden van een Jacobiaanmatrix van een kinematische ketting naar een van zijn gewrichtshoeken zijn originele symbolische formules ontwikkeld.

De toepasbaarheid van verscheidene van de theoretische resultaten is ook in *experimenten* met een industriële robot aangetoond. Voor deze experimenten volstaat een relatief eenvoudige robotcontroleprogrammatuur.

Samenvatting

Kinematische modellen en onzekerheidsidentificatie voor krachtgestuurde robottaken

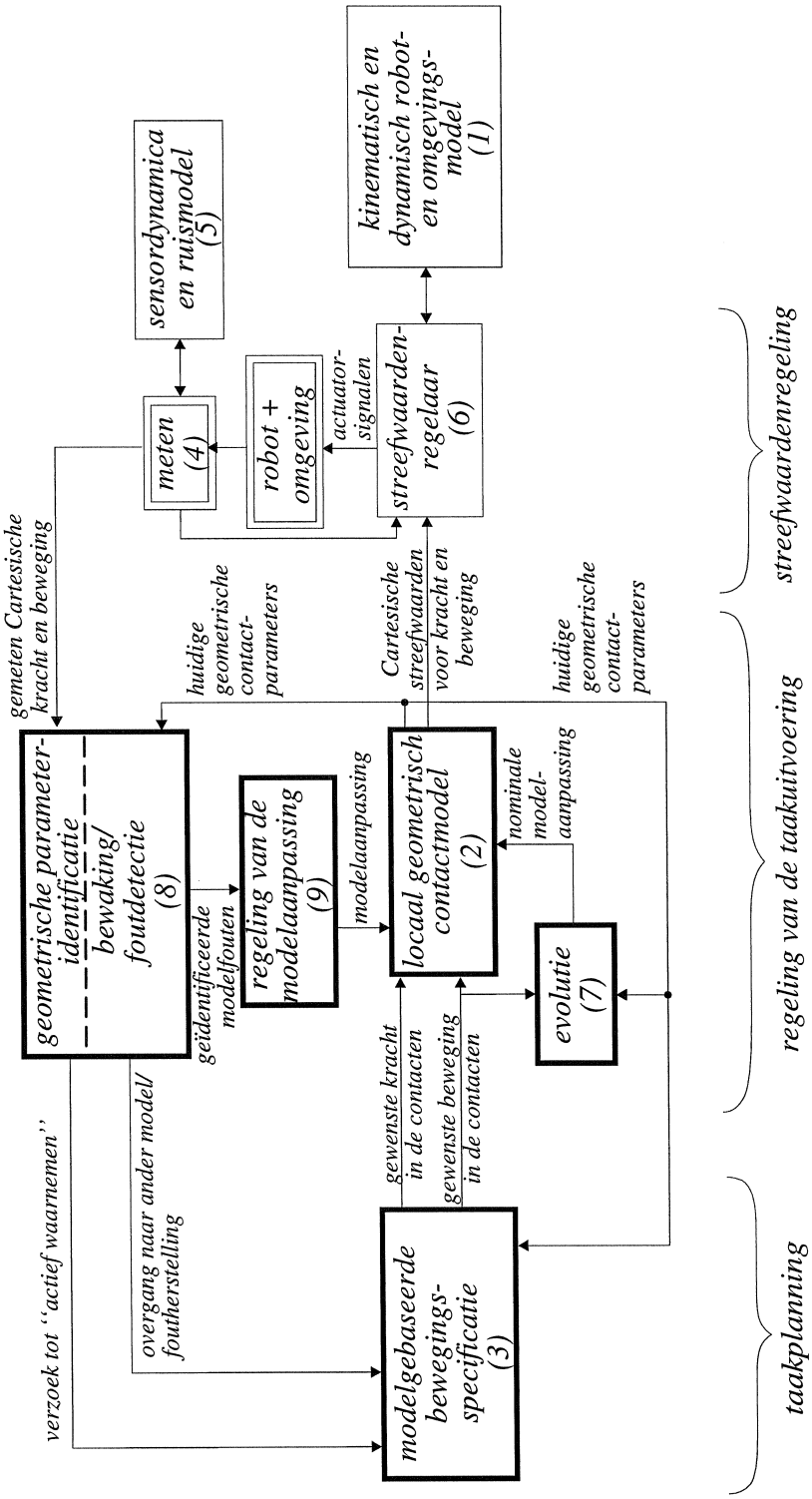
1 Inleiding

Deze thesis behandelt krachtgestuurde robottaken, waarbij de bewegingsbeperkingen die aan de robot worden opgelegd, variëren tijdens de taakuitvoering. Niet de regeltechnische aspecten worden benadrukt, maar wel : 1) het modelleren van de geometrie van een contactsituatie; 2) het modelleren van onzekerheden in het geometrisch model; en 3) het ontwerpen van identificatievergelijkingen voor het schatten van deze onzekerheden, gebaseerd op de ogenblikkelijke kracht- en bewegingsmetingen. Deze drie aspecten van een algemeen krachtsturings-systeem worden geïntegreerd binnen het unificerende kader van de kinematica van seriële manipulatoren.

Componenten van een krachtsturings-systeem

Een krachtgestuurde robot manipuleert objecten terwijl deze in contact zijn met de omgeving. Deze interacties met de omgeving (OMG) beperken de bewegingsvrijheid van het gemanipuleerde object (OBJ), en genereren reactiekrachten. De taaksturing moet op een veilige en voorspelbare manier met deze reactiekrachten omgaan, en wel op de wijze die de gebruiker gespecifiëerd heeft. FIGUUR 1 schetst de verschillende programmatuurmodules die tesamen de robot het gewenste gedrag moeten bezorgen. (Het weze opgemerkt dat een volledig systeem zoals in FIG. 1 nog nergens operationeel is, ook niet voor andere dan krachtsensoren.

Modelleren Dit is de belangrijkste stap in het ontwerp van een computer-gestuurd systeem : enkel de sensoren en de motoren komen in aanraking met de werkelijke wereld;



FIGUUR 1 : *Overzicht van een algemeen regelsysteem voor krachtgestuurde taken.* De modules met een vette omkadering worden dieper uitgewerkt in deze tekst. De modules met een dubbele rand bevatten fysische objecten; de andere modules bestaan uitsluitend uit programmatuur.

alle andere componenten bevatten enkel programmatuur. De kracht van een computergestuurd systeem wordt dus in hoge mate bepaald door de kwaliteit van de wiskundige modellen waarop het gebaseerd is. In het algemeen beschikt de stuurcomputer over de volgende modellen :

1. Een kinematisch en dynamisch model van de robot en van zijn interactie met de omgeving.

2. Een geometrisch model van de ogenblikkelijke bewegingsvrijheid van het *OBJ*.

3. Een model dat de dynamica en het ruisgedrag beschrijft die de sensoren in het totale systeem introduceren.

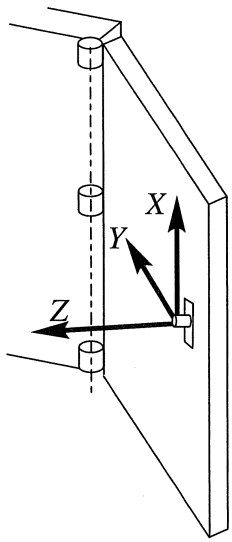
Deze tekst behandelt het geometrische model. Dit model bevat : 1) *eerste-orde* parameters (de positie van het contactpunt, en de richting van de contactnormaal), en 2) *tweede-orde* parameters (de kromming van de oppervlakken van *OBJ* en *OMG* in de onmiddellijke buurt van het contactpunt).

Bewegingsspecificatie Deze module gebruikt het locale geometrische model om ogenblikkelijke streefwaarden te genereren voor de beweging van het *OBJ*, en voor de gewenste reactiekrachten.

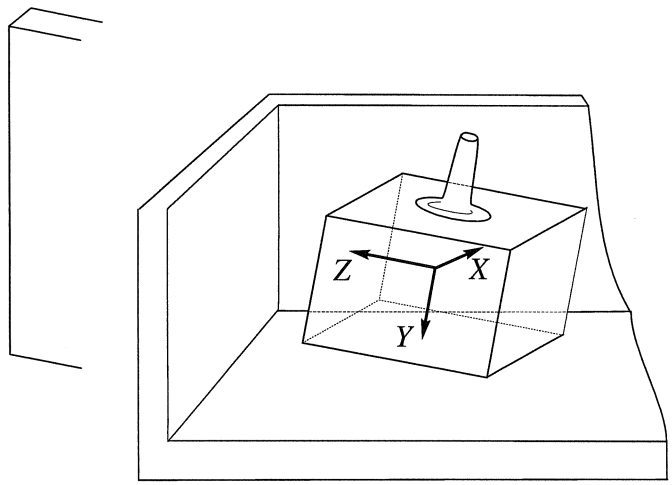
Meten Deze tekst veronderstelt ruisloze metingen, en een volledig stijve robot en sensoren (zodat vervormingen in het mechanische gedeelte van het systeem geen storingen introduceren in de gemeten bewegingen).

Regelen De regeltechnische aspecten van het probleem zijn in de literatuur reeds ten overvloede aan bod gekomen. In die mate zelfs dat de indruk gewekt wordt dat de uiteindelijke performantie van het systeem enkel een kwestie is van de goede regelwet te implementeren. In het algemeen gebruikt de regelaar zowel *terugkoppeling* als *voorwaartse koppeling*. Deze tekst bespreekt een manier om goede voorwaarste koppeling te genereren, door middel van een bewegingsspecificatie die optimaal gebruikt maakt van de beschikbare informatie over de locale geometrie van het contact. Dynamische invloeden worden niet beschouwd.

Evolutie, Identificatie en Modelaanpassing De regelaar hoeft minder terugkoppeling te gebruiken als de voorwaartse koppeling nauwkeurig de veranderingen in de locale geometrie kan volgen. Dit volggedrag wordt verzekerd door de gezamelijke inspanningen van drie modules : 1) *Evolutie*, t.t.z. de voorspelling van hoe het model verandert, op basis van het model zelf en van de *gespecifiëerde* beweging; 2) *Identificatie*, t.t.z. het schatten van de fouten in het model, op basis van het model zelf en de *gemeten* beweging en reactiekrachten; 3) *Modelaanpassing*, op basis van de geschatte afwijkingen en volgens een bepaalde regelwet. In sommige gevallen geeft de nominaal gespecifiëerde beweging niet



FIGUUR 2 :
Openen van een deur.



FIGUUR 3 : *Een blok in een hoek plaatsen.*

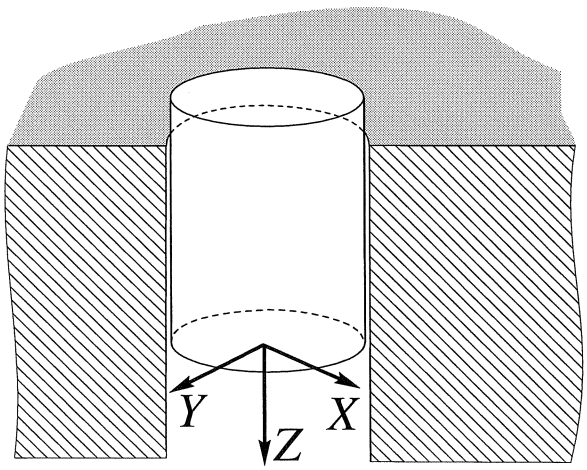
voldoende informatie om alle onzekerheden te identificeren. De taakplanner superponeert dan "aktieve waarnemingsbewegingen" bovenop het nominale traject, om die onzekerheden te "exciteren."

Bewaking Deze module houdt de overeenkomst in het oog tussen enerzijds het model, en anderzijds de geschatte modelfouten en de gemeten krachten en bewegingen. Als de afwijkingen te groot worden signaleert de bewakingsmodule dit aan de taakplanner. Die schakelt dan over op een nieuw geometrisch model en de bijhorende bewegingsspecificatie.

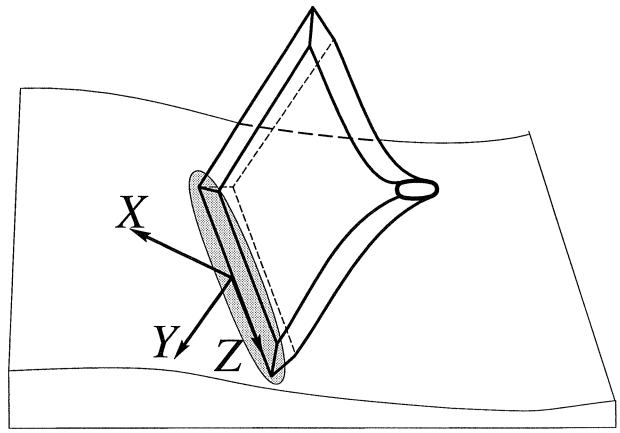
Taakspecificatie : het Taakassenstelsel-formalisme

De literatuur heeft het zogenaamde *taakassenstelsel* aanvaardt als een geschikte manier om een krachtgestuurde taak te specifiëren en te regelen. Dit gaat als volgt : 1) de gebruiker plaats een rechthoekig assenstelsel op een geschikte plaats op het *OBJ* of in de *OMG*; 2) hij geeft gewenste snelheden *of* krachten op langs elk van de assen (d.w.z., *zes* streefwaarden in totaal, want elke as wordt gebruikt in translatie en oriëntatie); en 3) hij specifiëert een eindconditie voor de taak. FIGUREN 2-6 geven enkele klassieke voorbeelden.

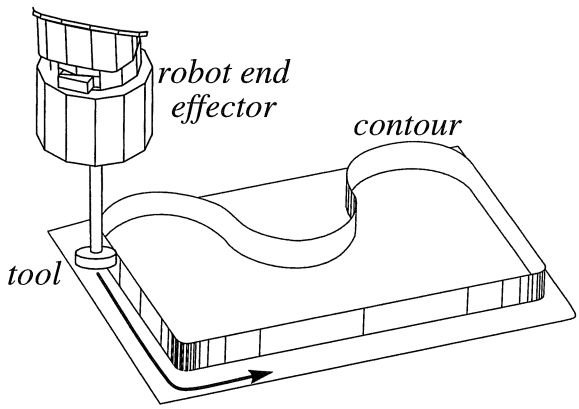
Het schatten van de ogenblikkelijke contactnormaal op basis van de gemeten krachten en/of bewegingen is de basis voor de robuustheid van de taakuitvoering tegen onzekerheden in de modellen. FIGUUR 7 illustreert het principe : de boogtangens van de verhouding tussen de kracht- of bewegingscomponenten volgens de gemodelleerde normaalrichting en de gemodelleerde raaklijn geeft een schatting voor de hoekfout tussen de gemodelleerde normaalrichting en de werkelijke.



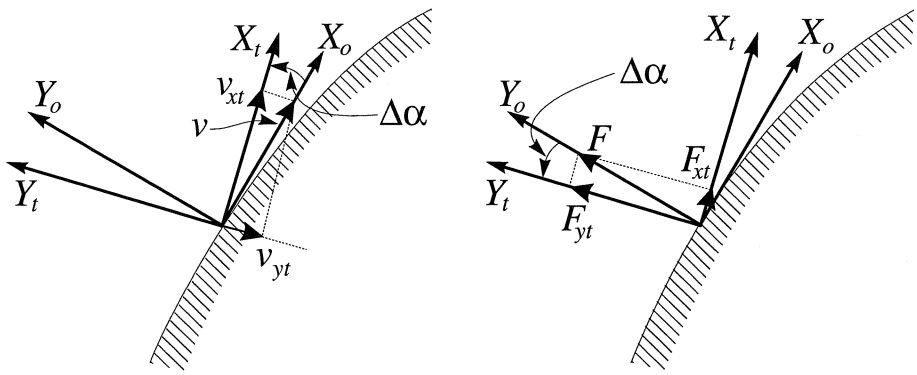
FIGUUR 4 : *Insertie van een pen.*



FIGUUR 5 : *Glijden over een oppervlak.*



FIGUUR 6 : *Volgen van een rand.*



FIGUUR 7 : *Schatten van de normaal.* Links : op basis van krachtmeting; rechts : op basis van snelheidsmeting.

Het Contactassenstelsel-formalisme

Het taakassenstelsel-formalisme heeft enkele beperkingen :

1. Het is niet in staat om alle mogelijke taken te specifiëren. Voorbeelden zijn contactsituaties waarin meer dan één contact optreedt, FIG. 8 en 9.

2. Het heeft geleid tot *niet-invariante* theorieën in verband met krachtsturing : dit betekent dat de resultaten afhangen van de keuze van de fysische eenheden en van het referentieassenstelsel. De oorzaak van dit probleem is het feit dat men het begrip van "orthogonaliteit" tussen de kracht- en bewegingsrichtingen heeft ingevoerd, terwijl dit geen fysische betekenis heeft voor krachten en bewegingen in de drie-dimensionale ruimte.

Deze tekst biedt oplossingen voor beide problemen :

1. Elk afzonderlijk contact wordt gemodelleerd met een *kinematische ketting* (de zogenaamde *virtuele contact-manipulator*) die het *OBJ* met de *OMG* verbindt op een manier die het *OBJ* dezelfde bewegingsbeperking oplegt als het echte contact. Voor contactsituaties met verscheidene contacten levert deze modellering een kinematisch systeem op met een aantal parallelle takken. De klassieke kinematica-literatuur beschrijft hoe men de bewegingen en/of reactiekrachten van zo'n complex systeem kan berekenen.

2. De nieuw ontwikkelde theoretische concepten in deze thesis zijn consistent beschreven op een invariante manier. Orthogonaliteit wordt vervangen door *reciprociteit*, wat wel een fysisch zinvol begrip is in de context van krachtgestuurde robots.

2 Jacobianen en hun afgeleiden

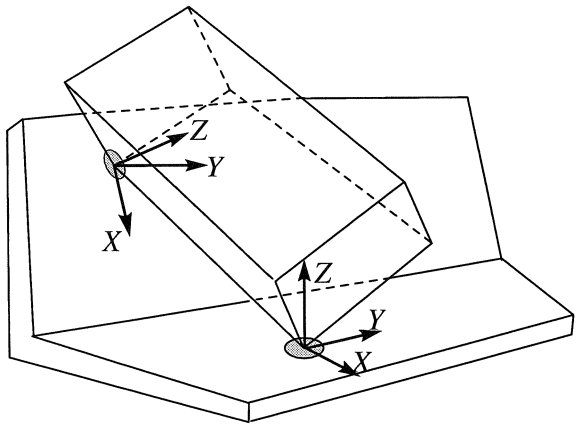
Deze sectie beschrijft de nieuwe bijdragen aan de berekening van de afgeleide van een Jacobiaan-matrix —van een willekeurige seriële kinematische ketting— naar een verandering in één van de gewrichtshoeken. Deze afgeleide wordt op een *symbolische* manier berekend als een *lineaire* functie van de Jacobiaan-matrix zelf.

Definities

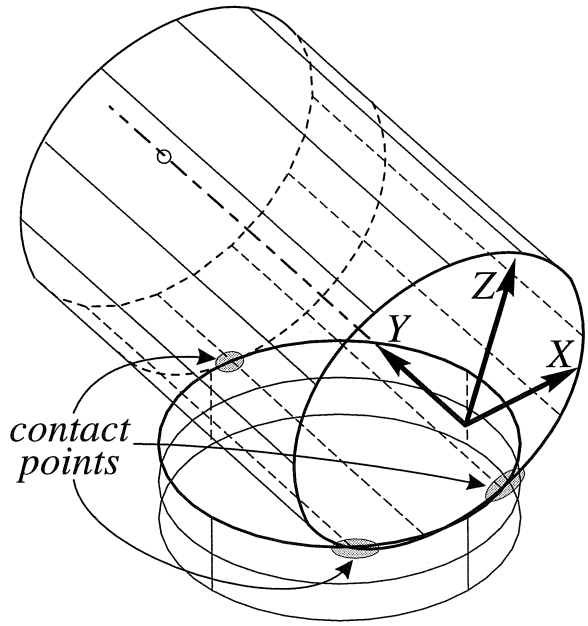
De ogenblikkelijke snelheid van een star lichaam is beschreven door een zes-vector t — de "*twist*"— die bestaat uit twee delen : $\mathsf{t} = [\boldsymbol{\omega}^T \ \boldsymbol{v}^T]^T$, met $\boldsymbol{\omega}$ de hoeksnelheid van het lichaam, en $\boldsymbol{v}$ zijn translatie-snelheid. Zowel $\boldsymbol{v}$ als $\boldsymbol{\omega}$ zijn drie-vectoren.

De ogenblikkelijke snelheid van het eindpunt van een seriële kinematische ketting met n gewrichten is een lineaire functie van de ogenblikkelijke gewrichtssnelheden $\dot{q}^i, i = 1, \ldots, n$:

$$\mathsf{t} = \begin{bmatrix} \boldsymbol{J}^1 & \ldots & \boldsymbol{J}^n \end{bmatrix} \begin{bmatrix} \dot{q}^1 \\ \vdots \\ \dot{q}^n \end{bmatrix} = \boldsymbol{J}\dot{\boldsymbol{q}},$$



FIGUUR 8 : *Niet-compatibel naadvolgen.*



FIGUUR 9 : *Pen-op-gat.*

met $\dot{\boldsymbol{q}} = [\dot{q}^1 \ldots \dot{q}^n]^T$.

De resultante van alle krachten die op een star lichaam inwerken is beschreven door een zes-vector w —de "*wrench*"— die bestaat uit twee delen : $\mathsf{w} = [\boldsymbol{f}^T \ \boldsymbol{m}^T]^T$ met $\boldsymbol{f}$ de lineaire component, en $\boldsymbol{m}$ de angulaire component ("moment" of "koppel"). Zowel $\boldsymbol{f}$ als $\boldsymbol{m}$ zijn drie-vectoren.

Krachten en snelheden transformeren op dezelfde wijze van een referentiestelsel $\{g\}$ naar een tweede stelsel $\{f\}$, FIG. 10 :

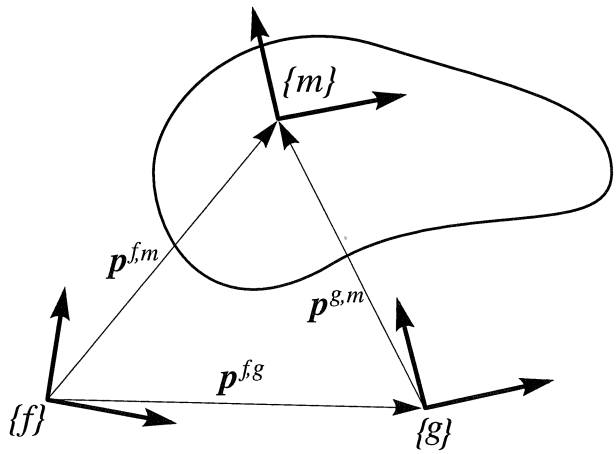
$$ {}_f\mathsf{t} = {}^g_f\boldsymbol{S} \ {}_g\mathsf{t} \qquad {}_f\mathsf{w} = {}^g_f\boldsymbol{S} \ {}_g\mathsf{w}. $$

${}^g_f\boldsymbol{S}$ is de zogenaamde *schroeftransformatie-matrix* :

$$ {}^g_f\boldsymbol{S} = \begin{bmatrix} {}^g_f\boldsymbol{R} & \boldsymbol{O}_{3\times 3} \\ \left[\boldsymbol{p}^{f,g}\times\right] {}^g_f\boldsymbol{R} & {}^g_f\boldsymbol{R} \end{bmatrix}. $$

${}^g_f\boldsymbol{R}$ is de drie-bij-drie rotatiematrix tussen $\{f\}$ en $\{g\}$. $\boldsymbol{O}_{3\times 3}$ is de drie-bij-drie nul-matrix. $\boldsymbol{p}^{f,g}$ is de vector van de oorsprong van $\{f\}$ naar de oorsprong van $\{g\}$; $\left[\boldsymbol{p}^{f,g}\times\right]$ is de scheefsymmetrische matrix

$$ \left[\boldsymbol{p}^{f,g}\times\right] = \begin{bmatrix} 0 & -p_z^{f,g} & p_y^{f,g} \\ p_z^{f,g} & 0 & -p_x^{f,g} \\ -p_y^{f,g} & p_x^{f,g} & 0 \end{bmatrix}. $$



FIGUUR 10 : *Geometrische parameters voor de transformatie van twists.* $\{f\}$ en $\{g\}$ zijn referentie-assenstelsels vast verbonden aan de wereld. De vector $\boldsymbol{p}^{i,j}$ verbindt de oorsprong van assenstelsel $\{i\}$ met de oorsprong van assenstelsel $\{j\}$.

Elke kolom van de Jacobiaan-matrix is gebaseerd op de geometrie van de seriële ketting, FIG. 11. Weze $\boldsymbol{e}^i$ de eenheidsvector volgens de rotatie-as van gewricht i, en laat de Z-as van een assenstelsel $\{i\}$ samenvallen met deze draairichting. Dan wordt de $i$de kolom van de Jacobiaan-matrix, uitgedrukt ten opzichte van het basis-assenstelsel $\{bs\}$:

$$\boldsymbol{J}^i = \begin{bmatrix} \boldsymbol{e}^i \\ \boldsymbol{p}^{bs,i} \times \boldsymbol{e}^i \end{bmatrix} = {}^{i}_{bs}\boldsymbol{S} \begin{bmatrix} \boldsymbol{k} \\ \mathbf{o} \end{bmatrix}, \qquad \boldsymbol{k} \triangleq \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Afgeleide naar een gewrichtshoek

Uit de voorgaande definities kunnen de volgende partiële afgeleiden berekend worden ($\{ee\}$ is het eindpunts-assenstelsel van de ketting) :

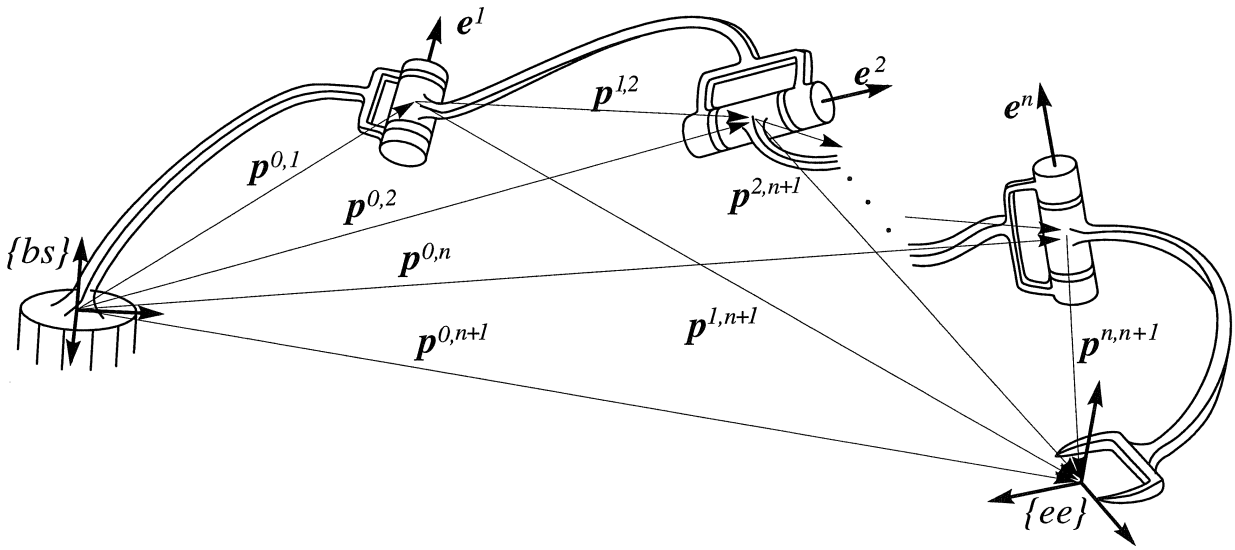
$$\frac{\partial^{i}_{bs}\boldsymbol{R}}{\partial q^j} = \left[\boldsymbol{e}^j \times\right] {}^{i}_{bs}\boldsymbol{R},$$

$$\frac{\partial^{ee}_{bs}\boldsymbol{S}}{\partial q^j} = \boldsymbol{S}_\Delta\left(\boldsymbol{J}^j\right) {}^{ee}_{bs}\boldsymbol{S}, \quad \text{met} \quad \boldsymbol{S}_\Delta\left(\boldsymbol{J}^j\right) \triangleq \begin{bmatrix} [\boldsymbol{e}^j\times] & \boldsymbol{O}_{3\times3} \\ [\boldsymbol{v}^j\times] & [\boldsymbol{e}^j\times] \end{bmatrix}, \quad \text{en} \quad \boldsymbol{v}^j = \boldsymbol{p}^j \times \boldsymbol{e}^j.$$

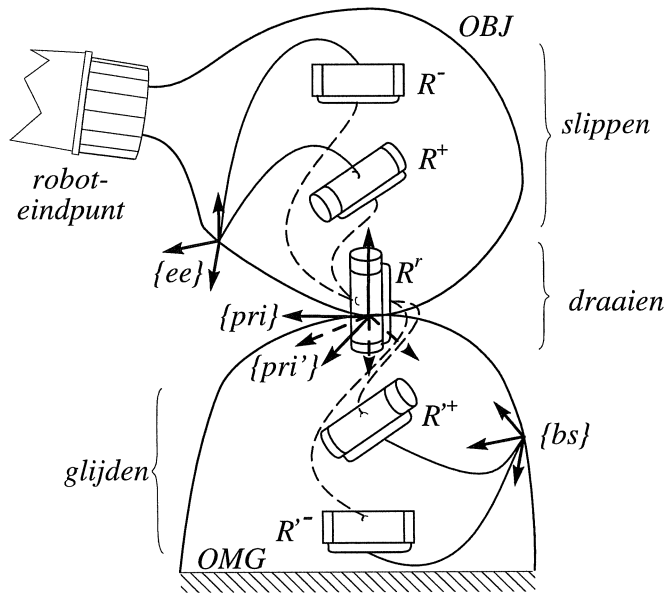
En uiteindelijk

$$\frac{\partial \boldsymbol{J}^i}{\partial q^j} = \boldsymbol{S}_\Delta\left(\boldsymbol{J}^j\right)\ \boldsymbol{J}^i.$$

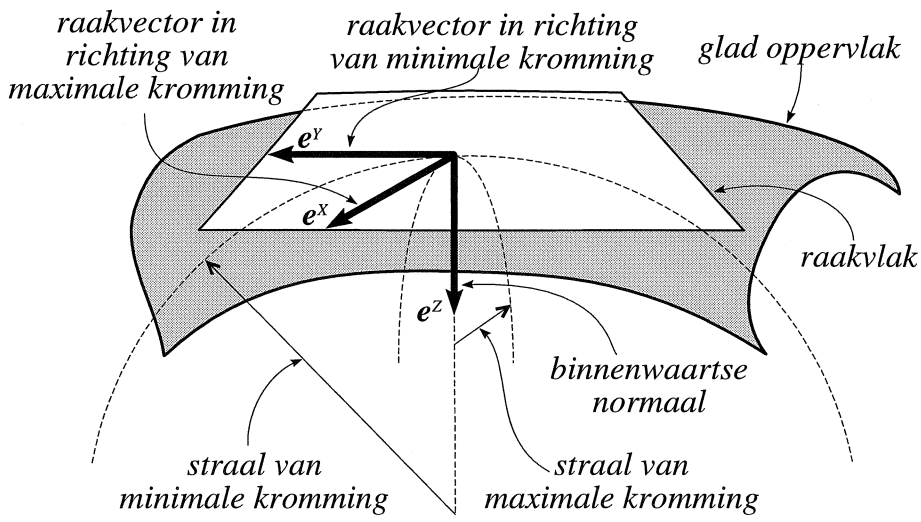
Merk op dat de bovenstaande afgeleiden niet mogen toegepast worden als $j \geq i$, of als de gewrichten i and j niet tot dezelfde kinematische ketting behoren. In deze gevallen worden de afgeleiden *identiek nul.*



FIGUUR 11 : *Geometrische parameters van een seriële kinematische ketting.*



FIGUUR 12 : *Virtuele manipulator voor een punt-contact tussen gladde oppervlakken.*



FIGUUR 13 : *Het hoofdkrommings-assenstelsel voor een glad oppervlak.*

3 Kinematische modellen

Deze sectie geeft een methode, gebaseerd op de theorie van kinematische systemen, om de ogenblikkelijke vrijheidsgraden te beschrijven van het gemanipuleeerde object, als het onderworpen is aan een willekeurige combinatie van bewegingsbeperkingen.

Punt-contact tussen gladde oppervlakken

Het punt-contact (of *oppervlak/oppervlak*-contact) is de meest algemene enkelvoudige contactsituatie. Alle andere kunnen door middel van limietovergangen hiervan afgeleid worden (zie verder). Het punt-contact laat vijf vrijheidsgraden toe. Deze vijf vrijheidsgraden worden gemodelleerd met een *virtuele contact-manipulator*, FIG. 12. Deze verbindt het *OBJ* met de *OMG* door een aantal stijve gelederen en rotatiegewrichten, die het *OBJ* dezelfde vijf vrijheidsgraden bieden als het fysische contact.

De manipulator is opgebouwd vanuit de locale tweede-orde geometrie van de oppervlakken in contact, FIG. 13 :

1. Eén rotatiegewricht, R^r, van de manipulator ligt langs de contactnormaal. De vrijheidsgraad van dit gewricht wordt *draaien* genoemd.

2. Een tweede, R^-, en een derde, R^+, liggen evenwijdig aan de richtingen van, respectievelijk, maximale en minimale kromming van het *OBJ*, en op afstanden van het contactpunt gelijk aan, respectievelijk, de stralen van minimale (r^-) en maximale (r^+) kromming. De X-as van het *hoofdkrommings-assenstelsel* $\{pri\}$ is de richting van maximale kromming; de Y-as is de richting van minimale kromming; en de Z-as is de richting van de contact-normaal.

   De vrijheidsgraden van deze gewrichten worden *slippen* genoemd.

3. Een gelijkaardige geometrische constructie bestaat voor de *OMG*. De notaties voor de geometrische parameters krijgen een accent.

   De vrijheidsgraden van deze gewrichten worden *glijden* genoemd.

De totale virtuele manipulator verbindt dus een "basis"-assenstelsel $\{bs\}$ op de *OMG* met een "eindpunts"-assenstelsel $\{ee\}$ op het *OBJ*. Merk op dat de virtuele manipulator *geen* echte seriële kinematische ketting is : de *slip*-gewrichten zijn onderling niet verbonden; de *glij*-gewrichten ook niet. Dit heeft echter geen enkele weerslag op de Jacobiaan-matrix : voor het beschrijven van de ogenblikkelijke snelheden die het *OBJ* kan uitvoeren in de gemodelleerde contactsituatie zijn enkel de ogenblikkelijke posities van de gewrichten van belang, niet hun onderlinge verbindingsstructuur.

Het is nu eenvoudig om de Jacobiaan-matrix $\boldsymbol{J}$ van de ogenblikkelijke bewegingsvrijheidsgraden op te stellen, alsook de Jacobiaan-matrix $\boldsymbol{G}$ van de ideale reactiekrachten. Ten opzichte van de hoofdkrommings-assenstelsels $\{pri\}$ en $\{pri'\}$ zijn de *glij*-, *draai*- en *slip*-vrijheidsgraden als volgt :

$$ {}_{pri'}\boldsymbol{J}^{glij} = \left[\begin{array}{cc} {}_{pri'}\boldsymbol{J}'^{+} & {}_{pri'}\boldsymbol{J}'^{-} \end{array} \right] = \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ -r'^{+} & 0 \\ 0 & r'^{-} \\ 0 & 0 \end{array} \right], $$

$$ {}_{pri}\boldsymbol{G} = {}_{pri}\boldsymbol{J}^{r} = \left[\begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right], \quad {}_{pri}\boldsymbol{J}^{slip} = \left[\begin{array}{cc} {}_{pri}\boldsymbol{J}^{+} & {}_{pri}\boldsymbol{J}^{-} \end{array} \right] = \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ -r^{+} & 0 \\ 0 & r^{-} \\ 0 & 0 \end{array} \right]. $$

Ten opzichte van een willekeurig assenstelsel $\{n\}$ wordt dit :

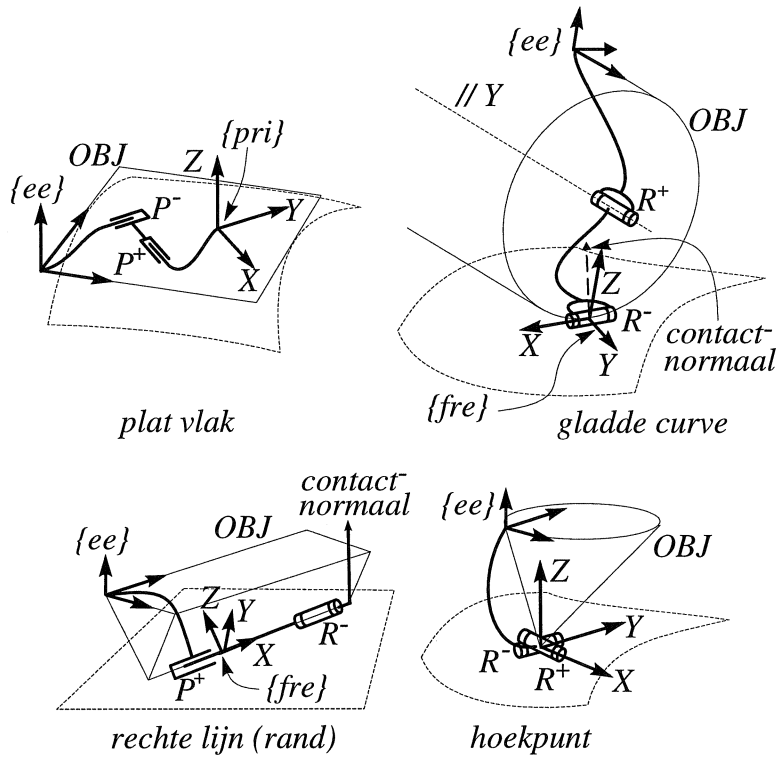
$$ {}_{n}\boldsymbol{J} = \left[{}^{pri'}_{n}\boldsymbol{S}\ {}_{pri'}\boldsymbol{J}^{glij} \quad {}^{pri}_{n}\boldsymbol{S}\ {}_{pri}\boldsymbol{J}^{r} \quad {}^{pri}_{n}\boldsymbol{S}\ {}_{pri}\boldsymbol{J}^{slip} \right], \quad {}_{n}\boldsymbol{G} = {}^{pri}_{n}\boldsymbol{S}\ {}_{pri}\boldsymbol{G}. $$

Bewegingsspecificatie

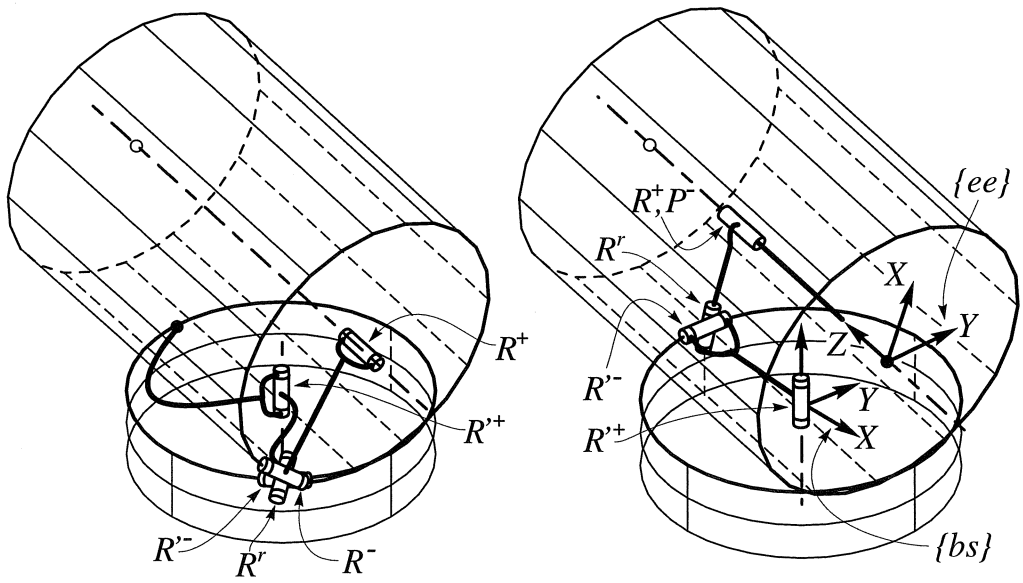
Om het *OBJ* een beweging te laten beschrijven die compatibel is met het contact, volstaat het om aan ieder van de vijf gewrichten een gewenste snelheid aan te leggen. Deze "gewrichtsruimte"-specificatie heeft enkele belangrijke voordelen :

1. De gegenereerde beweging is compatibel met het contact *door constructie*.
2. Bewegen over gekromde oppervlakken vereist een *versnelling* van het *OBJ*. Door de virtuele manipulator wordt die versnelling correct gegenereerd, ook al legt men aan de gewrichten slechts *snelheden* op.

De bovenstaande methode werkt echter slechts correct indien het model van de contactsituatie correct is.



FIGUUR 14 : *Kinematische modellen voor de verschillende types contact-oppervlakken.*



FIGUUR 15 : *Pen-op-gat : kinematische modellen.* Links : *kromme/kromme*-contact; rechts : *oppervlak/kromme*-contact. Alle gewrichten zijn draaiparen, behalve het schuifpaar "P^-" langs de as van de pen.

Andere contact-types

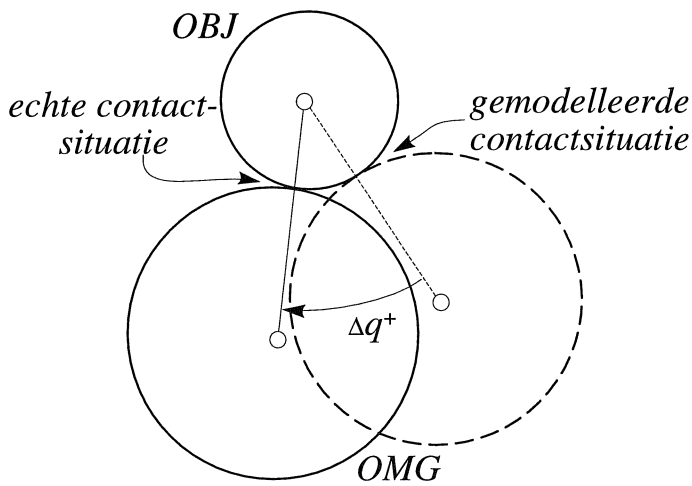
Door de krommingsstraal van het *OBJ* en/of de *OMG* naar nul of oneindig te laten gaan bekomt men de andere enkelvoudige contact-types, FIG. 14. De Jacobiaan-matrices worden even eenvoudig afgeleid. Merk op dat in sommige limietgevallen het aantal vrijheidsgraden vermindert. Dit is ook dadelijk te merken aan de rang van de Jacobiaan-matrices.

Meervoudige contacten

Wanneer meer dan één contact op het *OBJ* ingrijpt, kan nog steeds elk van deze contacten afzonderlijk gemodelleerd worden met een virtuele contact-manipulator zoals in de vorige paragrafen. De *pen-op-gat* contact-situatie is een goed voorbeeld, FIG. 15. Het *OBJ* hangt nu vast aan de omgeving met een aantal kinematische kettingen in parallel. De klassieke kinematica biedt hulpmiddelen aan om voor zo'n meer complex systeem de bewegingsvrijheidsgraden en de mogelijke ideale reactiekrachten te berekenen, en om de gewenste beweging van het *OBJ* te specifiëren :

1. Veronderstel dat n_c contacten op het *OBJ* aangrijpen. De Jacobiaan-matrices $\boldsymbol{J}^i$ en de gewrichtssnelheden $\dot{\boldsymbol{q}}^i$ zijn dan gekoppeld door de volgende *kringloopvergelijkingen* :

$$\exists\, \dot{\boldsymbol{q}}^1, \ldots, \dot{\boldsymbol{q}}^{n_c} : \mathsf{t} = \boldsymbol{J}^1\dot{\boldsymbol{q}}^1 = \ldots = \boldsymbol{J}^{n_c}\dot{\boldsymbol{q}}^{n_c}.$$



FIGUUR 16 : *Kromme/kromme-contact : tijdsvariante onzekerheid van eerste orde.* De onzekerheid wordt voorgesteld met een onzekerheid in een gewrichtshoek van de virtuele contactmanipulator.

Een equivalente uidrukkingswijze voor de resulterende totale bewegingsvrijheid is :

$$\boldsymbol{J}^{\text{totaal}} = \bigcap_{i=1}^{n_c} \boldsymbol{J}^i.$$

2. Terwijl de bewegingsvrijheden van het *OBJ* afnemen naarmate er meer contacten bijkomen, groeit de vectorruimte van ideale reactiekrachten die door die contacten kunnen gegenereerd worden :

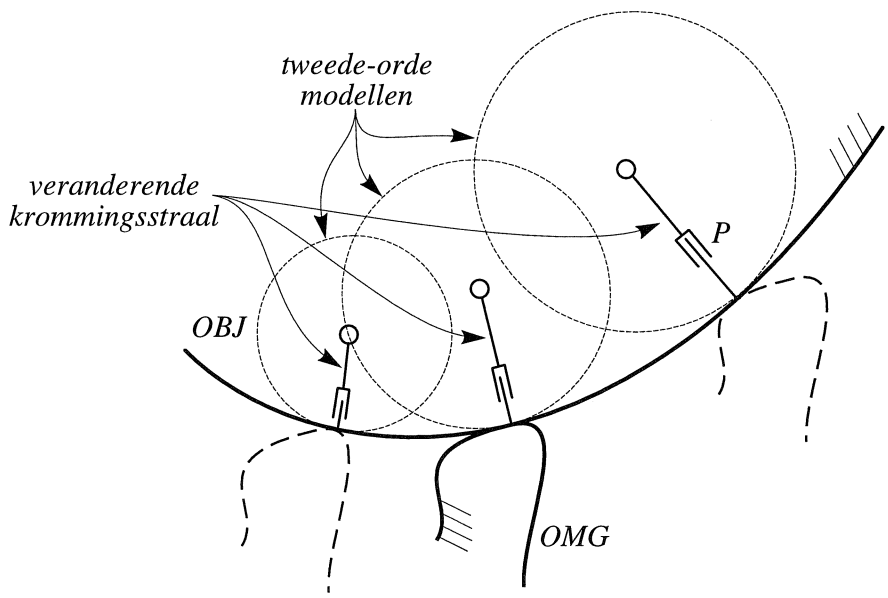
$$\text{span}\left(\boldsymbol{G}^{\text{totaal}}\right) = \text{span}\left(\left[\boldsymbol{G}^1 \ \ldots \ \boldsymbol{G}^{n_c}\right]\right), \quad \text{or} \quad \boldsymbol{G}^{\text{totaal}} = \bigoplus_{i=1}^{n_c} \boldsymbol{G}^i.$$

Als de rang van $\boldsymbol{J}^{\text{totaal}}$ gelijk is aan n_f, dan is de rang van $\boldsymbol{G}^{\text{totaal}}$ gelijk aan $6 - n_f$.

Bewegingsspecificatie In principe verloopt de bewegingsspecificatie voor een meervoudig contact op dezelfde wijze als voor een enkelvoudig contact : leg een gewenste snelheid aan aan de aandrijvende gewrichten. Het verschil is nu dat voor een meervoudig contact niet alle gewrichten onafhankelijk kunnen aangedreven worden. De kringloopvergelijkingen uit de vorige paragraaf helpen de gebruiker om uit de n_c verschillende contacten n_f onafhankelijke gewrichten te kiezen. Voor de *pen-op-gat* contact-situatie van FIG. 15 zijn dit de R^+, R'^+ en R'^- gewrichten van het *oppervlak/kromme*-contact.

4 Onzekerheden

Het is niet realistisch om te verwachten dat men de taakregelaar steeds van perfecte geometrische modellen kan voorzien. Er bestaan steeds onzekerheden in de geometrische parameters van het model. Deze sectie beschrijft hoe deze geometrische onzekerheden in de



FIGUUR 17 : *Kromme/kromme-contact : tijdsvariante onzekerheid van tweede orde.* De onzekerheid wordt voorgesteld met een prismatisch gewricht P, dat niet tot de virtuele contactmanipulator behoort.

virtuele contact manipulator kunnen geïntegreerd worden. Twee types van *tijdsvariërende* onzekerheden worden beschouwd : 1) onzekerheden van *eerste orde* (d.w.z. onzekerheden in de kennis van de positie van het contactpunt en de richting van de contactnormaal; zie FIG. 16 voor een 2D voorbeeld); en 2) onzekerheden van *tweede orde* (d.w.z. onzekerheden in de hoofdkrommingsrichtingen en de waarde van de hoofdkrommingen, zie bijvoorbeeld FIG. 17).

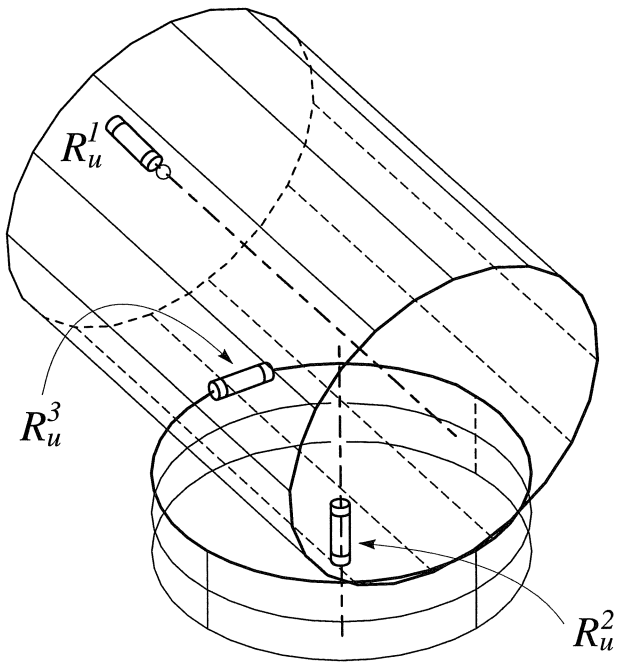
De eerste-orde onzekerheden zijn eenvoudig te integreren : zij stemmen overeen met onnauwkeurigheden in de kennis van de gewrichtshoeken van de virtuele manipulator.

Tweede-orde onzekerheden vragen een uitbreiding van het kinematische model : de veranderingen in de kromtestralen van het *OBJ* worden gemodelleerd door een schuifpaar te plaatsen tussen elk van de *slip*-gewrichten en het *draai*-gewricht; voor de *OMG* is de aanpak volledig gelijkaardig.

Onzekerheden verstoren de nominale Jacobiaan-matrix. De invloed van een onzekerheid υ ("upsilon") is formeel te schrijven als een Taylor-reeksontwikkeling in de parameter υ :

$$\boldsymbol{J}(\boldsymbol{\Upsilon}) = \boldsymbol{J}(\boldsymbol{\Upsilon} = 0) + \sum_{i=1}^{n_u} \upsilon_i \frac{\partial \boldsymbol{J}}{\partial \upsilon_i},$$

$$\dot{\boldsymbol{J}}(\boldsymbol{\Upsilon}) = \dot{\boldsymbol{J}}(\boldsymbol{\Upsilon} = 0) + \sum_{i=1}^{n_u} \upsilon_i \frac{\partial \dot{\boldsymbol{J}}}{\partial \upsilon_i},$$



FIGUUR 18 : *Pen-op-gat : minimum aantal onzekerheidsgewrichten.*

$$\boldsymbol{G}(\upsilon) = \boldsymbol{G}(\upsilon = 0) + \sum_{i=1}^{n_u} \upsilon_i \frac{\partial \boldsymbol{G}}{\partial \upsilon_i}.$$

Veronderstel dat er n_u^1 eerste-orde onzekerheden zijn —$\mathcal{U}^1 = \{\upsilon_1^1, \ldots, \upsilon_1^{n_u^1}\}$— en n_u^2 tweede-orde onzekerheden —$\mathcal{U}^2 = \{\upsilon_1^2, \ldots, \upsilon_2^{n_u^2}\}$— dan geldt :

$$\boldsymbol{J}(\mathcal{U}^1, \mathcal{U}^2) \;=\; \boldsymbol{J}(0,0) + \sum_{j=1}^{n_u^1} \frac{\partial \boldsymbol{J}}{\partial \upsilon_1^j}\, \upsilon_1^j + \sum_{j=1}^{n_u^2} \frac{\partial \boldsymbol{J}}{\partial \upsilon_2^j}\, \upsilon_2^j,$$

$$\dot{\boldsymbol{J}}(\mathcal{U}^1, \mathcal{U}^2) = \dot{\boldsymbol{J}}(0,0) + \sum_{j=1}^{n_u^1} \left\{ \frac{\partial \boldsymbol{J}}{\partial \upsilon_1^j}\, \dot{\upsilon}^{1,j} + \frac{\partial \dot{\boldsymbol{J}}}{\partial \upsilon_1^j}\, \upsilon_1^j \right\} + \sum_{j=1}^{n_u^2} \left\{ \frac{\partial \boldsymbol{J}}{\partial \upsilon_2^j}\, \dot{\upsilon}^{2,j} + \frac{\partial \dot{\boldsymbol{J}}}{\partial \upsilon_2^j}\, \upsilon_2^j \right\}.$$

$\boldsymbol{J}(0,0)$ en $\dot{\boldsymbol{J}}(0,0)$ zijn de snelheids-Jacobiaan en zijn tijdsafgeleide, zoals beschreven in het huidige model.

Eén van de hoofdredenen om robottaken in contact met de omgeving uit te voeren is dat de contacten gebruikt worden om het *OBJ* te leiden naar een gewenste positie. Dit betekent intuïtief dat er minder onzekerheidsparameters zouden moeten zijn bij meervoudige contacten. Dit is inderdaad zo, maar enkel indien de verschillende virtuele contact-manipulatoren *dezelfde* basis- en eindpunts-assenstelsels verbinden. Een mooi voorbeeld hiervan is *pen-op-gat* : FIG. 18 toont de drie onzekerheidsgewrichten, die overblijven na reductie van al de overbodige onzekerheidsgewrichten.

5 Identificatie

Tijdens de taakuitvoering gebruikt de regelaar het geometrische contactmodel zowel om een zo goed mogelijke beweging te specifiëren, als om de reactiekrachten te regelen. Het model echter is onderhevig aan voortdurende wijzigingen, aangezien de contactgeometrie verandert als het *OBJ* beweegt over de *OMG*. Dit wil zeggen dat de gespecifiëerde beweging en de verwachte reactiekrachten afwijken ten opzichte van de werkelijk gemeten waarden. Deze afwijkingen kunnen echter op hun beurt gebruikt worden om te schatten waar en hoeveel het model onnauwkeurig is.

Reciprociteit en consistentie

Deze tekst gebruikt twee complementaire wegen om *identificatievergelijkingen* op te stellen voor alle gemodelleerde eerste- en tweede-orde onzekerheden. Een eerste aanpak steunt op het *reciprociteits*-beginsel : een snelheid $\mathsf{t} = [\boldsymbol{v}^T \ \boldsymbol{\omega}^T]^T$ is reciprook aan een kracht $\mathsf{w} = [\boldsymbol{f}^T \ \boldsymbol{m}^T]^T$ als t geen vermogen levert tegen w. Of :

$$\boldsymbol{m}^T\boldsymbol{\omega} + \boldsymbol{f}^T\boldsymbol{v} = 0, \text{ of } \mathsf{t}^T\tilde{\boldsymbol{\Delta}}\mathsf{w} = 0, \quad \text{met} \quad \tilde{\boldsymbol{\Delta}} = \begin{bmatrix} \boldsymbol{O}_{3\times3} & \boldsymbol{I}_{3\times3} \\ \boldsymbol{I}_{3\times3} & \boldsymbol{O}_{3\times3} \end{bmatrix}.$$

De tijdsafgeleide hiervan geeft een tweede-orde reciprociteits-voorwaarde :

$$\frac{\partial}{\partial t}\left(\boldsymbol{m}^T\boldsymbol{\omega} + \boldsymbol{f}^T\boldsymbol{v}\right) = 0, \quad \text{of} \quad \frac{\partial}{\partial t}\left(\mathsf{t}^T\tilde{\boldsymbol{\Delta}}\mathsf{w}\right) = 0.$$

Een tweede weg gebruikt de *consistentie*-voorwaarde, die uitdrukt dat de gemodelleerde en de gemeten bewegingsparameters compatibel moeten zijn :

$$\forall i = 1, \ldots n_c, \exists \dot{\boldsymbol{q}}^i = \left[\dot{q}^i_1 \ldots \dot{q}^i_{n^f_i}\right]^T \ : \mathsf{t}_m = \boldsymbol{J}^i\dot{\boldsymbol{q}}^i,$$

$$\exists \boldsymbol{\mathcal{T}} = \left[\boldsymbol{\mathcal{T}}^1 \ldots \boldsymbol{\mathcal{T}}^{n_c}\right]^T, \boldsymbol{\mathcal{T}}^i = [\tau^i_1 \ldots \tau^i_{6-n^f_i}]^T \ : \mathsf{w}_m = \boldsymbol{G}\boldsymbol{\mathcal{T}}.$$

n_c is het aantal parallelle contacten. n^f_i is het aantal bewegingsvrijheidsgraden in contact i. $\boldsymbol{J}^i$ is de Jacobiaan-matrix van het $i$de contact. $\dot{q}^i_j$ is de snelheid van het $j$de gewricht in de $i$de virtuele contact-manipulator. $\boldsymbol{G}$ is kracht-Jacobiaan-matrix van de totale bewegingsbeperking, d.w.z. de aaneenschakeling van alle $\boldsymbol{G}^i$: $\boldsymbol{G} = \left[\boldsymbol{G}^1 \ \ldots \ \boldsymbol{G}^{n_c}\right]$. $\boldsymbol{\mathcal{T}}^i = [\tau^i_1 \ldots \tau^i_{6-n^f_i}]^T$ is de vector van de reactiekrachten in bewegingsbeperking i.

Consistentie- en reciprociteits-gebaseerde identificatie

De hoger beschreven Taylorreeks-ontwikkelingen van de Jacobiaan-matrices leiden tot de volgende identificatie-vergelijkingen :

Consistentie :

1. Eerste-orde :

$$\mathsf{t}_m = \left\{ \boldsymbol{J}^i(\mathcal{V}^i = 0) + \sum_{j=1}^{n_u^i} \frac{\partial \boldsymbol{J}^i}{\partial v_j^i} v_j^i \right\} \dot{\boldsymbol{q}}^i, \qquad \forall i = 1, \ldots, n_c,$$

$$\mathsf{w}_m = \left\{ \boldsymbol{G}(\mathcal{V} = 0) + \sum_{v_j \in \mathcal{V}} \frac{\partial \boldsymbol{G}}{\partial v_j} v_j \right\} \boldsymbol{\tau}.$$

2. Tweede-orde :

$$\dot{\mathsf{t}}_m = \frac{\partial \left(\boldsymbol{J}^i(\mathcal{V}^i)\ \dot{\boldsymbol{q}}^i \right)}{\partial t} = \dot{\boldsymbol{J}}^i(\mathcal{V}^i)\ \dot{\boldsymbol{q}}^i + \boldsymbol{J}^i(\mathcal{V}^i)\ \ddot{\boldsymbol{q}}^i.$$

Reciprociteit :

1. Eerste-orde :

$$\begin{aligned} \mathsf{t}_m^T\ \tilde{\boldsymbol{\Delta}}\ \boldsymbol{G}^i(\mathcal{V}^i = 0) \ &+ \ \sum_{j=1}^{n_u^i} \mathsf{t}_m^T\ \tilde{\boldsymbol{\Delta}}\ \frac{\partial \boldsymbol{G}^i}{\partial v_j^i} v_j^i \ = \ 0, \\ \mathsf{w}_m^T\ \tilde{\boldsymbol{\Delta}}\ \boldsymbol{J}(\mathcal{V} = 0) \ &+ \ \sum_{v_j \in \mathcal{V}} \mathsf{w}_m^T\ \tilde{\boldsymbol{\Delta}}\ \frac{\partial \boldsymbol{J}}{\partial v_j} v_j \ = \ 0. \end{aligned}$$

2. Tweede-orde :

$$0 = \frac{\partial \left(\mathsf{t}_m^T\ \tilde{\boldsymbol{\Delta}}\ \boldsymbol{G}^i(\mathcal{V}) \right)}{\partial t} = \dot{\mathsf{t}}_m^T\ \tilde{\boldsymbol{\Delta}}\ \boldsymbol{G}^i(\mathcal{V}) + \mathsf{t}_m^T\ \tilde{\boldsymbol{\Delta}}\ \dot{\boldsymbol{G}}^i(\mathcal{V}).$$

De equivalentie tussen de identificatie-vergelijkingen bekomen op basis van reciprociteit en op basis van consistentie kan bewezen worden.

Identificatie in de praktijk

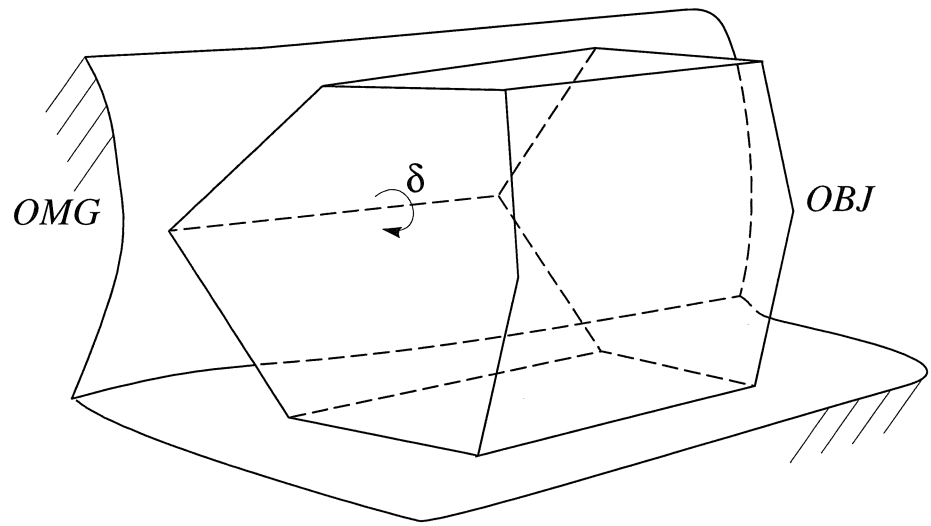
De onderstaande tabel geeft een vergelijkend overzicht tussen beide identificatie-wegen.

Reciprociteit	**Consistentie**
De identificatie-vergelijkingen zijn *lineair* in de onzekerheidsparameters.	De identificatie-vergelijkingen zijn *niet-lineair* in de onzekerheidsparameters, want ook de virtuele gewrichtssnelheden of de individuele contactkrachten treden als onbekenden op.
$\dim(\boldsymbol{J})$ *kracht-gebaseerde* vergelijkingen voor de *totale* bewegingsbeperking.	6 *kracht-gebaseerde* vergelijkingen voor de *totale* bewegingsbeperking.
$6-n_f^i$ *snelheids-gebaseerde* vergelijkingen voor contact i.	6 *snelheids-gebaseerde* vergelijkingen voor contact i.
Identificatie en modelaanpassing : 1. Bereken de onzekerheden uit de lineaire vergelijkingen. 2. Pas het model aan, d.w.z. dat het nu compatibel is met de metingen. 3. Los de sluitingsvergelijkingen op voor de virtuele gewrichtssnelheden en de individuele contactkrachten.	Identificatie en modelaanpassing : 1. Los de niet-lineaire vergelijkingen op naar de onbekenden, d.w.z. de virtuele gewrichtssnelheden, de individuele contactkrachten en de onzekerheden. Het oplossen van een niet-lineair systeem gebruikt, in het algemeen, een aantal iteratiestappen. In elk van deze tussenstappen is het model (nog) niet compatibel met de metingen; d.w.z. dat de gewrichtssnelheden en de reactiekrachten niet ondubbelzinnig kunnen opgelost worden. 2. Pas het model aan.

De identificatie-vergelijkingen zijn afgeleid in de veronderstelling van een ideale wereld. In werkelijkheid spelen echter enkele theoretische en praktische problemen een rol :

Theoretische problemen

1. De coëfficiënt van een onzekerheid v kan *identiek nul* zijn. De onzekerheid is dan *niet identificeerbaar.*
2. De coëfficiënt van een onzekerheid v kan ogenblikkelijk nul worden, door een toevallige combinatie van kracht- en/of snelheidsmetingen. v is dan *voorwaardelijk niet identificeerbaar.*
3. Het stelsel identificatie-vergelijkingen kan *over-* of *onderbepaald* zijn. Dit bemoeilijkt het ogenblikkelijk oplossen ervan.



FIGUUR 19 : *Voorbeeld van een niet-identificeerbare onzekerheid.* De contact-normaal van de linkse rand van het object kan niet worden geëxciteerd omdat het *vlak/vlak*-contact op de bodem van het object de vereiste bewegingen verhindert.

4. Bij meervoudige contacten wordt de kracht-gebaseerde identificatie bemoeilijkt door het feit dat de gemeten reactiekracht niet meer ondubbelzinnig kan ontbonden worden in bijdragen van de verschillende individuele contacten.

5. Bij meervoudige contacten wordt de snelheids-gebaseerde identificatie bemoeilijkt door het feit dat sommige onzekerheden niet meer *geëxciteerd* kunnen worden; zie bijvoorbeeld FIG. 19

Praktische problemen

1. *Wrijving* verstoort de krachtmetingen.

2. *Soepelheid* in de mechanische structuur verstoort de snelheidsmeting.

3. *Sensorruis* verstoort alle metingen.

4. Industriële robots zijn geen heel nauwkeurige meettoestellen. Bijgevolg kunnen contactsituaties met kleine afmetingen niet geïdentificeerd worden. Bijvoorbeeld : complexe spuitgietstukken, geïntegreerde schakelingen,...

Voorbeeld : contacten tussen veelvlakken

De algemene theoretische identificatie-vergelijkingen zijn toegepast op een aantal veelvuldig voorkomende contactsituaties, zoals het tweedimensionaal randvolgen, en de contacten tussen veelhoeken.

2D randvolgen

Sommige identificatie-vergelijkingen voor de eerste-orde onzekerheden waren reeds uit de literatuur bekend. Alle mogelijke andere kunnen met de in deze thesis voorgestelde identificatie-methoden afgeleid worden. Voor tweede-orde onzekerheden bestond nog geen literatuur. De volgende interessante tweede-orde identificatie-vergelijking is met bovenvermelde reciprociteits-gebaseerde technieken gevonden :

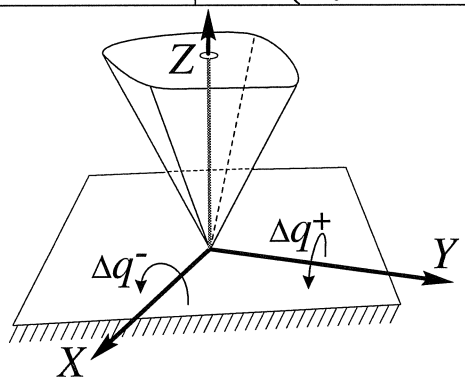
$$\frac{\Delta r}{r} = \frac{a}{v\omega}.$$

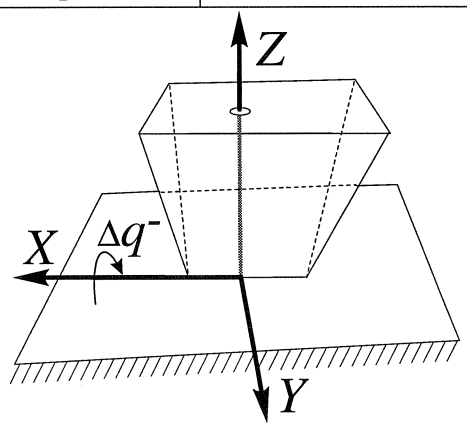
r is de huidige gemodelleerde waarde van de kromtestraal van de OMG. Δr is de geschatte afwijking op deze kromtestraal. a is het verschil tussen de versnelling van het gemodelleerde contactpunt en de reëel gemeten versnelling van het contactpunt. v is de ogenblikkelijke tangentiële snelheid van het contactpunt. ω is de ogenblikkelijke rotatiesnelheid van de contactnormaal.

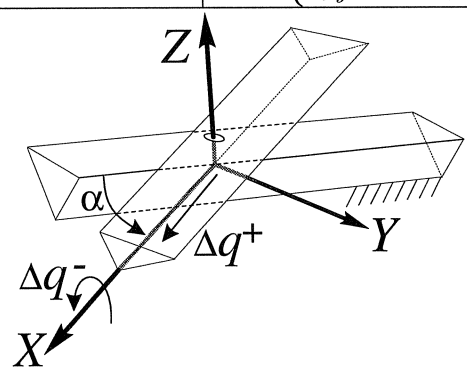
Veelvlakken

Deze paragraaf somt de identificatie- en "bewakings"-vergelijkingen op die met de bovenvermelde identificatie-aanpak werden afgeleid. De bewakingsvergelijkingen worden gevonden uit de consistentie-gebaseerde identificatie : het zijn betrekkingen tussen de gemeten grootheden, die identiek nul zouden moeten zijn indien het geometrische model overeenstemt met de werkelijkheid. Als deze overeenstemming niet meer opgaat is het gebruikte type van model waarschijnlijk niet meer geldig.

De onzekerheidsparameters worden in de bijhorende figuren schematisch aangeduid.

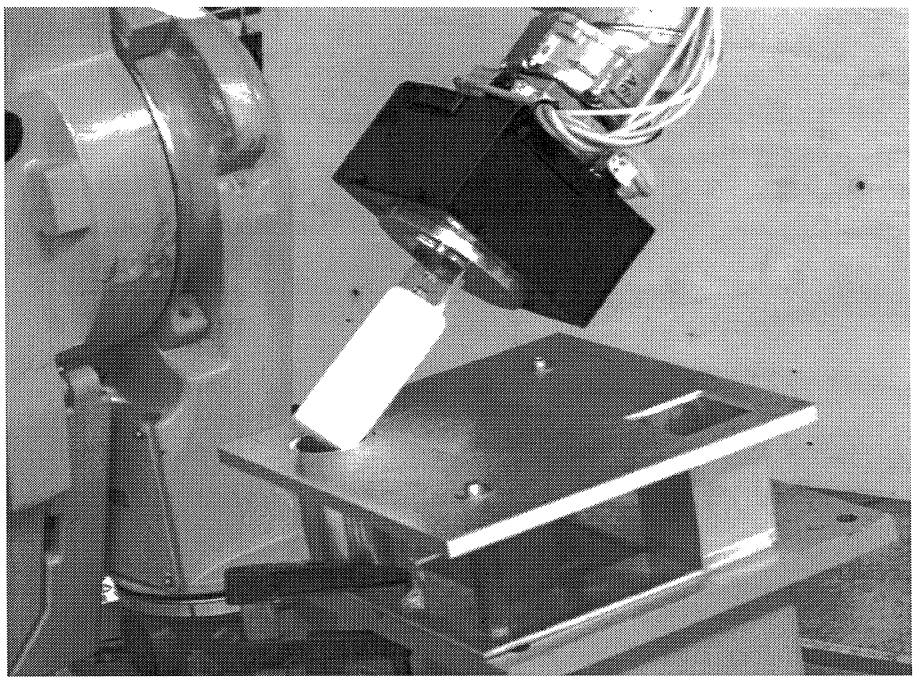
<table>
<tr><td colspan="2">Punt/vlak</td></tr>
<tr><td>beweging</td><td>kracht</td></tr>
<tr><td colspan="2">identificatie</td></tr>
<tr><td>$v_z + \Delta q^+ v_x - \Delta q^- v_y = 0$</td><td>$\begin{cases} f_x - \Delta q^+ f_z = 0 \\ f_y + \Delta q^- f_z = 0 \end{cases}$</td></tr>
<tr><td colspan="2">
</td></tr>
<tr><td colspan="2">bewaking</td></tr>
<tr><td>$v_z = 0$</td><td>$f_x = f_y = m_x = m_y = m_z = 0$</td></tr>
</table>

Rand/vlak	
beweging	kracht
identificatie	
$\begin{cases} v_z - \Delta q^- v_y = 0 \\ \omega_y + \Delta q^- \omega_z \end{cases}$	$f_y + \Delta q^- f_z = 0$
	
bewaking	
$v_z = \omega_y = 0$	$f_x = f_y = m_x = m_z = 0$

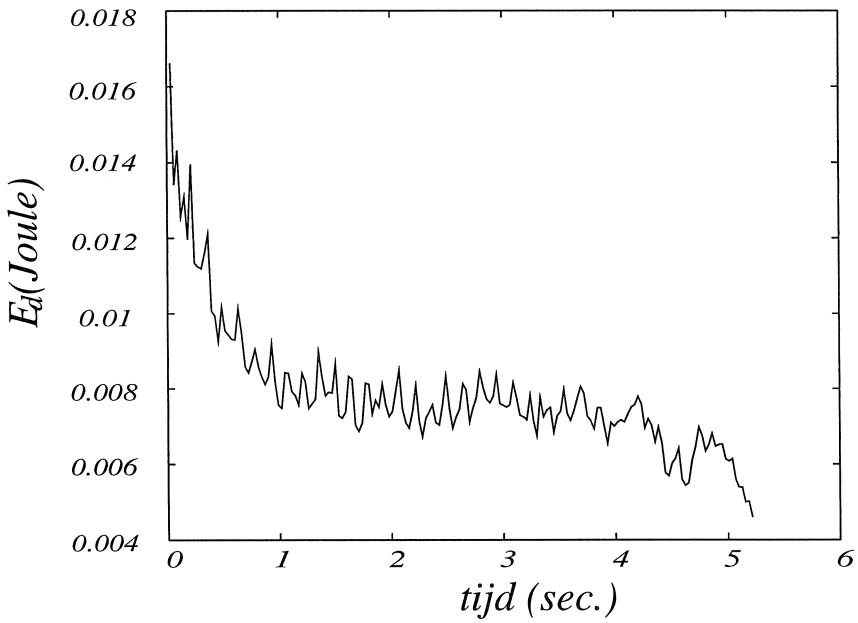
Rand/rand	
beweging	kracht
identificatie	
$v_z - \Delta q^- v_y - \Delta q^+ \omega_y = 0$	$\begin{cases} m_y + \Delta q^+ f_z = 0 \\ f_y + \Delta q^- f_z = 0 \end{cases}$
	
bewaking	
$v_z = 0$	$f_x = f_y = m_x = m_y = m_z = 0$

6 Experimenten

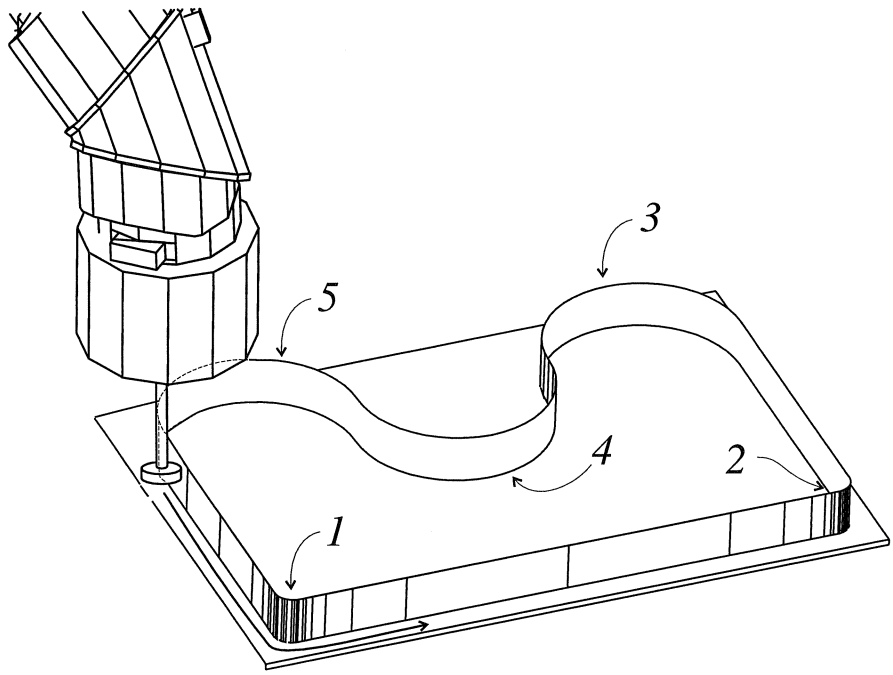
Deze sectie illustreert enkele van de theoretische ontwikkelingen beschreven in de vorige secties aan de hand van werkelijk uitgevoerde experimenten.



FIGUUR 20 : *Pen-op-gat : experimentele opstelling.* (Figuur welwillend ter beschikking gesteld door S. Demey en Adhi Sudadi.)



FIGUUR 21 : *Pen-op-gat : vervormingsenergie van het soepele element gedurende de aligneerbeweging van de pen.* De initiële hoekfout bedroeg ongeveer 60 graden, en de aligneersnelheid was constant $-0.2\ rad/sec.$ (Figuur welwillend ter beschikking gesteld door S. Dutré.)



FIGUUR 22 : *Volgen van een vlakke naad : CAD-model.*

Pen-op-gat

De *pen-op-gat*-taak, FIG. 20, is een voorbeeld van een taak met meervoudige, tijdsvariërende contacten. Het kinematisch model is gebruikt om de beweging te specifiëren waarbij de pen van een zeer slecht gealigneerde positie (hoekfout van ongeveer 60 graden!) rechtgedraaid wordt, zonder verlies van de drie contactpunten, FIG. 15. Tijdens deze aligneerbeweging meet men de vervormingsenergie, die door de soepele krachtsensor wordt opgeslagen, FIG. 21. (Deze soepelheid vangt de kleine afwijkingen op die steeds bestaan tussen de gespecifiëerde en de werkelijk mogelijke bewegingen.) Het blijkt dat deze vervormingsenergie niet toeneemt, wat erop duidt dat het gebruikte model een zeer goede bewegingsspecificatie genereert. (De initiële "piek" in de vervormingsenergie is te wijten aan inertiële effecten : er wordt een constant snelheidsprofiel opgelegd; d.w.z. dat er een sprong in de versnelling optreedt, die een piek in de vervormingsenergie veroorzaakt.)

Volgen van een vlakke rand

De hogervermelde identificatie-formule voor een tweede-orde onzekerheid is toegepast tijdens het volgen van een cylindrische omgeving. TABEL 1 resumeert de resultaten. Naarmate het model reeds correcter is, is ook de schatting van de nog overblijvende fout beter.

gemodelleerde straal (mm)	geschatte Δr (mm)	totale straal (exact : $299mm$)
290	8	298
250	50	300
200	110	310

TABEL 1 : *2D randvolgen van een cylindrisch object : geschatte fout op de gemodelleerde straal.*

7 Algemeen besluit

De vorige secties presenteerden enkele theoretische bijdragen aan het verhogen van de "intelligentie" van krachtgestuurde robottaken. Het bereikte intelligentieniveau blijft echter nog steeds zeer bescheiden, zowel in deze thesis als wereldwijd voor alle andere onderzoeksprojecten van deze aard. Achteraf bekeken kan men misschien beter spreken over het verhogen van de *kennis over* het uitvoeren van robottaken waarin contacten met de omgeving een hoofdrol spelen. Op zich is dit misschien het meest vruchtbare resultaat van dit onderzoek : vroeger was krachtsturing een monolitisch blok regeltechniek, waarbij men contactmodellering, bewegingsspecificatie en identificatie van onzekerheden hoogstens als als deelproblemen beschouwde met voor de hand liggende oplossingen; deze thesis draagt ertoe bij dat dit monolitisch blok wordt opgesplitst in interagerende deelprocessen, ieder met een welomschreven, beperkte verantwoordelijkheid, én met zijn eigen theoretische onderbouw. Het zoeken naar verbeteringen aan de bestaande systemen kan dus meer gericht en gelocaliseerd gebeuren. De volgende paragrafen geven een overzicht van de belangrijkste theoretische bijdragen van het gepresenteerde onderzoek.

Belangrijkste bijdragen

Modelleren van bewegingsbeperkingen. Op basis van de locale eerste- en tweede-orde geometrie van het gemanipuleerde object en de omgeving is een kinematisch model opgesteld van de bewegingsbeperkingen opgelegd door een algemeen *oppervlak/oppervlak*-contact. De modellen voor andere contact-types worden afgeleid als limietgevallen hiervan. De bewegingsspecificatie en de wiskundige beschrijving van het kinematisch model lopen volledig parallel aan wat gebruikelijk is voor mechanische robotarmen. Onzekerheden in de geometrie van het contactmodel worden eenvoudig in het model geïntegreerd.

Model-gebaseerde bewegingsspecificatie. De *slip*-, *draai*- en *glij*-vrijheidsgraden beschrijven perfect de reciproke beweging van het gemanipuleerde object. Het volstaat zelfs om deze vrijheidsgraden met een eerste-orde specificatie aan te drijven om een tweede-orde beweging voor het object te genereren. De uitbreiding naar meervou-

dige contacten gebeurt op basis van de *sluitingsvergelijkingen* uit de kinematica van complexe systemen.

Identificatie van geometrische onzekerheden. Op basis van de fysische voorwaarden voor reciprociteit en consistentie zijn identificatie-vergelijkingen afgeleid, die een verband leggen tussen, enerzijds, het huidige geometrische model van de bewegingsbeperking, en de gemeten krachten en snelheden, en, anderzijds, de fouten op de geometrische parameters die niet met zekerheid gekend zijn.

Manipulator-kinematica. Nieuwe symbolische formules zijn afgeleid voor de berekening van de partiële afgeleide van een Jacobiaan-matrix naar een verandering in één van de gewrichtshoeken.

Symbols and definitions

$\triangleq$: the left-hand side is a shorthand notation for the expression on the right-hand side

a : scalar (unbold lowercase)

$\boldsymbol{a}$: column vector (bold lowercase)

$\boldsymbol{A}$: matrix (bold uppercase)

$\boldsymbol{a}^{old}, \boldsymbol{J}^{slip}$: unbold lowercase trailing superscript indicates characteristic feature

$\boldsymbol{A}_{n\times m}$: n-by-m matrix (lowercase unbold subscript), e.g. :

$$\boldsymbol{I}_{3\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{O}_{3\times 2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$\boldsymbol{a}^i, \boldsymbol{A}^i$: ith vector or matrix in a set (index is unbold lowercase trailing superscript)

$a_i, \boldsymbol{A}_i$: ith element of vector $\boldsymbol{a}$ or ith column in matrix $\boldsymbol{A}$ (index is unbold lowercase trailing subscript); e.g.:

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad \boldsymbol{A} = [\boldsymbol{A}_1 \ \boldsymbol{A}_2 \ \ldots \ \boldsymbol{A}_m]$$

a_x, a_y, a_z : components of three-vector with respect to a reference frame

A_{ij} : element of matrix $\boldsymbol{A}$ on row i and column j (index is unbold trailing subscript):

$$\boldsymbol{A}_{n\times m} = \begin{bmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nm} \end{bmatrix}$$

$\boldsymbol{a}^T, \boldsymbol{A}^T$: transpose (unbold uppercase T as trailing superscript)

$\boldsymbol{a}^T\boldsymbol{b}$: scalar product

$\boldsymbol{AB}$: matrix product

$\boldsymbol{p} \times \boldsymbol{r}$: cross product of three-vectors

$[\boldsymbol{p}\times]$: matrix representing cross product with vector $\boldsymbol{p}$:

$$[\boldsymbol{p}\times] = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}$$

$\boldsymbol{A}^\dagger$: pseudo-inverse of matrix $\boldsymbol{A}$

$\boldsymbol{A}_K^\dagger$: weighted pseudo-inverse of matrix $\boldsymbol{A}$, with $\boldsymbol{K}$ as weighting matrix on the range space of $\boldsymbol{A}$

$\dot{\boldsymbol{a}}, \dot{\boldsymbol{A}}$: time derivative of a vector $\boldsymbol{a}$ or matrix $\boldsymbol{A}$

$\widetilde{\boldsymbol{\Delta}}$: 6×6 reciprocity matrix:

$$\widetilde{\boldsymbol{\Delta}} = \begin{bmatrix} \boldsymbol{O}_{3\times3} & \boldsymbol{I}_{3\times3} \\ \boldsymbol{I}_{3\times3} & \boldsymbol{O}_{3\times3} \end{bmatrix}$$

$\mathbb{R}$: set of real numbers

$\{a, b, c\}$: set containing elements a, b and c

MO : manipulated object (object rigidly connected to robot end effector)

ENV : environment

$\{ref\}$: reference frame (unbold lowercase, between curly braces)

$\{con\}$: contact frame

$\{pri\}$: principal frame on MO (smooth surface)

$\{pri'\}$: principal frame on ENV (smooth surface)

$\{fre\}$: Frenet frame on MO (smooth curve)

$\{fre'\}$: Frenet frame on ENV (smooth curve)

${}_n a, {}_{ref}\boldsymbol{a}, {}_o\boldsymbol{A}$: indication of reference frame in which scalar, vector or matrix is expressed (unbold leading subscript)

$\boldsymbol{p}$: position vector

$\boldsymbol{p}^{n,o}$: position vector, pointing from n to o

X, Y, Z : coordinate axes of right-handed orthogonal reference frame

$\boldsymbol{e}$: unit direction vector:

$$\boldsymbol{e} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}$$

$\boldsymbol{e}^X, \boldsymbol{e}^Y, \boldsymbol{e}^Z$: unit vectors along coordinate axes of a reference frame

$\boldsymbol{e}^{X,o}, \boldsymbol{e}^{Y,o}, \boldsymbol{e}^{Z,o}$: unit vectors along coordinate axes of the reference frame $\{o\}$

$\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$: coordinate vectors of unit vectors along coordinate axes

$$\boldsymbol{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \ \boldsymbol{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ \boldsymbol{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

R : revolute joint

P : prismatic joint

r, r^+, r^- : radius of curvature, radius of maximum curvature, radius of minimum curvature

$\kappa, \kappa^+, \kappa^-$: curvature $\kappa = 1/r, \kappa^+ = 1/r^+, \kappa^- = 1/r^-$

R^+, R^- : revolute joint at radius of maximum, respectively minimum, curvature

r'^+, r'^-, R'^+, R'^- : as r^+, r^-, R^+ and R^-, but for the *ENV* instead of the *MO*

P^+, P^-, P'^+, P'^- : prismatic joints along the corresponding radii of curvature

$\boldsymbol{\omega}$: angular velocity three-vector:

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

$\dot{\boldsymbol{\omega}}$: angular acceleration three-vector

$\boldsymbol{v}$: translational (or rectilinear) velocity three-vector:

$$\boldsymbol{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

t : time

t : velocity twist of a rigid body or reference frame:

$$\mathsf{t} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}$$

$\mathsf{t}^{ee}, \mathsf{t}^{bs}$: twist of the end effector frame of a a manipulator with the origin of the end effector frame, respectively the origin of the base frame, chosen as the reference point for the translational velocity component of the twist

${}_{ee}\mathsf{t}$, ${}_{bs}\mathsf{t}$: twist of the end effector frame of a a manipulator with its components expressed in the end effector reference frame, or the base reference frame, respectively.

t_Δ : infinitesimal displacement twist (roll δ_x, pitch δ_y, yaw δ_z, translation $[d_x\ d_y\ d_z]^T$):

$$\mathsf{t}_\Delta = \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ d_x \\ d_y \\ d_z \end{bmatrix}$$

$\boldsymbol{f}$: force:

$$\boldsymbol{f} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

$\boldsymbol{m}$: moment (torque):

$$\boldsymbol{m} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$$

w : wrench (generalized force) on rigid body:

$$\mathsf{w} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{m} \end{bmatrix}$$

$\mathsf{t}_m, \mathsf{w}_m$: measured twist and wrench

${}^o_n\boldsymbol{T}$: 4×4 homogeneous transformation matrix; maps coordinates of three-vectors from reference frame $\{o\}$ to reference frame $\{n\}$:

$${}^o_n\boldsymbol{T} = \begin{bmatrix} {}^o_n\boldsymbol{R} & \boldsymbol{p}^{n,o} \\ \boldsymbol{O}_{1\times 3} & 1 \end{bmatrix}.$$

$\boldsymbol{p}^{n,o}$ is the position vector of $\{o\}$'s origin in reference frame $\{n\}$.

${}^o_n\boldsymbol{R}$: 3×3 rotation matrix from reference frame $\{o\}$ to reference frame $\{n\}$:

$${}^o_n\boldsymbol{R} = \begin{bmatrix} {}_n\boldsymbol{e}^{X,o} & {}_n\boldsymbol{e}^{Y,o} & {}_n\boldsymbol{e}^{Z,o} \end{bmatrix}$$

${}^{o}_{n}\boldsymbol{T}_{\Delta}$: 4×4 infinitesimal homogeneous transformation matrix, corresponding to the infinitesimal displacement twist t_{Δ}:

$$ {}^{o}_{n}\boldsymbol{T}_{\Delta} = \begin{bmatrix} 1 & -\delta_z & \delta_y & d_x \\ \delta_z & 1 & -\delta_x & d_y \\ -\delta_y & \delta_x & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

${}^{o}_{n}\boldsymbol{S}$: 6×6 screw transformation matrix (transformation of twists and wrenches) from frame $\{o\}$ to frame $\{n\}$:

$$ {}_{n}\mathsf{t} = {}^{o}_{n}\boldsymbol{S}\ {}_{o}\mathsf{t}, \quad {}_{n}\mathsf{w} = {}^{o}_{n}\boldsymbol{S}\ {}_{o}\mathsf{w}, $$

$$ {}^{o}_{n}\boldsymbol{S} = \begin{bmatrix} {}^{o}_{n}\boldsymbol{R} & \boldsymbol{O}_{3\times 3} \\ [\boldsymbol{p}^{n,o}\times]\, {}^{o}_{n}\boldsymbol{R} & {}^{o}_{n}\boldsymbol{R} \end{bmatrix} $$

$\boldsymbol{S}_{\Delta}(\mathsf{t})$: 6×6 infinitesimal screw transformation matrix corresponding to the twist t:

$$ \boldsymbol{S}_{\Delta}(\mathsf{t}) = \boldsymbol{S}_{\Delta}\left(\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}\right) = \begin{bmatrix} [\boldsymbol{\omega}\times] & \boldsymbol{O}_{3\times 3} \\ [\boldsymbol{v}\times] & [\boldsymbol{\omega}\times] \end{bmatrix} $$

${}^{o}_{n}\boldsymbol{P}$: 6×6 screw projection matrix (projection of twist and wrench components) from frame $\{o\}$ to frame $\{n\}$:

$$ {}_{n}\mathsf{t} = {}^{o}_{n}\boldsymbol{P}\ {}_{o}\mathsf{t}, \quad {}_{n}\mathsf{w} = {}^{o}_{n}\boldsymbol{P}\ {}_{o}\mathsf{w}, $$

$$ {}^{o}_{n}\boldsymbol{P} = \begin{bmatrix} {}^{o}_{n}\boldsymbol{R} & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{O}_{3\times 3} & {}^{o}_{n}\boldsymbol{R} \end{bmatrix} $$

$\boldsymbol{P}_{\Delta}(\mathsf{t})$: 6×6 infinitesimal screw projection matrix corresponding to twist t:

$$ \boldsymbol{P}_{\Delta}(\mathsf{t}) = \boldsymbol{P}_{\Delta}\left(\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}\right) = \begin{bmatrix} [\boldsymbol{\omega}\times] & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{O}_{3\times 3} & [\boldsymbol{\omega}\times] \end{bmatrix} $$

$\boldsymbol{M}_o^n$: 6×6 reference point transformation matrix (transformation of twist and wrench representations if velocity reference point on the rigid body changes) from origin of frame $\{o\}$ to origin of frame $\{n\}$:

$$\mathsf{t}^n = \boldsymbol{M}_o^n \, \mathsf{t}^o \quad {}_n\mathsf{w} = \boldsymbol{M}_o^n \, {}_o\mathsf{w}$$

$$\boldsymbol{M}_o^n = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{O}_{3\times3} \\ [\boldsymbol{p}^{no}\times] & \boldsymbol{I}_{3\times3} \end{bmatrix}$$

$\boldsymbol{M}_\Delta(\mathsf{t})$: 6×6 infinitesimal reference point transformation matrix corresponding to twist t:

$$\boldsymbol{M}_\Delta(\mathsf{t}) = \boldsymbol{M}_\Delta\left(\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{O}_{3\times3} & \boldsymbol{O}_{3\times3} \\ [\boldsymbol{v}\times] & \boldsymbol{O}_{3\times3} \end{bmatrix}$$

q : generalized joint position

$\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$: column vector of generalized joint positions and its derivatives with respect to time:

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}, \; \dot{\boldsymbol{q}} = \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}, \; \ddot{\boldsymbol{q}} = \begin{bmatrix} \ddot{q}_1 \\ \vdots \\ \ddot{q}_n \end{bmatrix}$$

$\boldsymbol{J}$: twist Jacobian (matrix mapping joint velocities to Cartesian velocities):

$$\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = \boldsymbol{J}\dot{\boldsymbol{q}}$$

$\boldsymbol{J}^+, \boldsymbol{J}^-$: columns of twist Jacobian $\boldsymbol{J}$ corresponding to rotation about R^+ and R^-, respectively

$\boldsymbol{J}^{slip}$: $\boldsymbol{J}^{slip} = \begin{bmatrix} \boldsymbol{J}^+ & \boldsymbol{J}^- \end{bmatrix}$

$\boldsymbol{J}'^+, \boldsymbol{J}'^-$: columns of twist Jacobian $\boldsymbol{J}$ corresponding to rotation about R'^+ and R'^-, respectively

$\boldsymbol{J}^{slide}$: $\boldsymbol{J}^{slide} = \begin{bmatrix} \boldsymbol{J}'^+ & \boldsymbol{J}'^- \end{bmatrix}$

$\boldsymbol{J}^r$: rotation about contact normal

τ : generalized joint torque

$\boldsymbol{\tau}, \dot{\boldsymbol{\tau}}$: column vector of generalized joint torques and its time derivative:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix}, \ \dot{\boldsymbol{\tau}} = \begin{bmatrix} \dot{\tau}_1 \\ \vdots \\ \dot{\tau}_n \end{bmatrix}$$

$\boldsymbol{G}$: wrench Jacobian (matrix mapping reaction forces in each contact to resultant Cartesian wrench on the end effector):

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{m} \end{bmatrix} = \boldsymbol{G\tau}$$

$\mathcal{U} = \{\upsilon_1, \ldots, \upsilon_{n_u}\}$: uncertainty parameter set of a kinematic motion constraint model, with n_u uncertainty parameters υ_i. Pronunciation: "upsilon"

$\upsilon_j^1, \upsilon_j^2$: jth first, respectively second, order uncertainty parameter

$\boldsymbol{K}$: 6×6 stiffness matrix

$\boldsymbol{C}$: 6×6 damping matrix

$\boldsymbol{I}$: 6×6 generalized inertia matrix

Contents

Preface i

Abstract v

Samenvatting vii
1 Inleiding . . . vii
2 Jacobianen en hun afgeleiden . . . xii
3 Kinematische modellen . . . xvi
4 Onzekerheden . . . xx
5 Identificatie . . . xxiii
6 Experimenten . . . xxix
7 Algemeen besluit . . . xxxii

Symbols and definitions xxxiii

1 Introduction 1
1.1 Components of compliant motion task control . . . 1
1.2 Compliant motion tasks: the Task Frame Formalism . . . 6
1.3 The Contact Frame Formalism . . . 12
1.4 Overview of the text . . . 15
1.5 Conclusions . . . 17

2 Literature survey 19
2.1 Introduction . . . 19
2.2 Non-invariance of the Task Frame Formalism . . . 19
2.3 Geometric modelling of motion constraints . . . 20
2.4 Motion specification . . . 21
2.5 Modelling of geometric uncertainties . . . 23
2.6 Identification of uncertainties . . . 24
2.7 Conclusions . . . 27

3 Jacobians 29
3.1 Introduction . . . 29
3.2 Twists and wrenches . . . 30

3.3 Jacobian matrix for a serial chain 36
3.4 Derivative with respect to a joint angle 39
3.5 Numerical results 46
3.6 Conclusions 48

4 Kinematic models **49**
4.1 Introduction 49
4.2 Point contact between smooth surfaces 49
4.3 Other contact types 57
4.4 Multiple contacts 60
4.5 Conclusions 67

5 Uncertainties **69**
5.1 Introduction 69
5.2 Uncertainty types 70
5.3 Kinematic model for uncertainties 74
5.4 Mathematical model 75
5.5 Representation uncertainties 75
5.6 Multiple contacts 78
5.7 Conclusions 80

6 Identification **83**
6.1 Introduction 83
6.2 Reciprocity and consistency 84
6.3 Consistency based identification 85
6.4 Reciprocity based identification 86
6.5 Equivalence between Reciprocity and Consistency 88
6.6 Identification in practice 88
6.7 Example: 2D contour following 94
6.8 First order identification of polyhedral contacts 103
6.9 Conclusions 105

7 Experiments and applications **109**
7.1 Controller hardware and software 109
7.2 Sphere-on-sphere 111
7.3 Peg-on-hole 112
7.4 Incompatible 3D seam following 115
7.5 2D contour following 119
7.6 Redundancy resolution 121
7.7 Suggested potential applications 125
7.8 Conclusions 127

8 General conclusions **129**
8.1 Main contributions 129
8.2 Critical evaluation 131
8.3 Future work 133

Bibliography **137**

A Derivatives for the surface-surface contact **145**
A.1 Twist and wrench Jacobians 145
A.2 Inertial representation 146
A.3 Body-fixed representation 147
A.4 Hybrid representation 148

B Derivatives for the 2D curve-curve contact **151**
B.1 Hybrid representation 151
B.2 Inertial representation 155
B.3 Body-fixed representation 156

C Link with differential geometry **159**

D Peg-on-hole **163**
D.1 Position closure for the surface-curve contact 163
D.2 Position closure for the curve-curve contact 164
D.3 Total position closure 165
D.4 Jacobian for the surface-curve contact 165
D.5 Jacobians for the curve-curve-contacts 166
D.6 Velocity closure 167

E Incompatible 3D seam following **169**
E.1 Definitions of geometric parameters 169
E.2 Motion constraint model 172
E.3 Motion specification 174

Chapter 1

Introduction

This text is about compliant motion with time varying motion constraints. A typical task is surface following —e.g., inspection, polishing or welding of intricate and (partially) unknown 3D surfaces— with tools that can have all kinds of contact with the surfaces, and even several contacts in parallel. A typical device to execute such compliant motion tasks is a serial six degrees-of-freedom robot arm, whose controller uses sensor information from the robot's joint encoders and from a wrist force/torque sensor. The tool, also called the "manipulated object," is attached to the robot end effector.

Firstly, this introductory Chapter gives a high-level overview of the different modules in a compliant motion control system. This overview sketches the context for those components that are developed in more detail by the other Chapters of the text.

Secondly, the state-of-the-art task specification formalism —the so-called *Task Frame Formalism*— is discussed, because task specification is the module that is most affected by the developments described in this text.

The third Section then gives an outline of what new task specification primitives have been developed, together with the motivation to do so.

1.1 Components of compliant motion task control

In a compliant motion task a force controlled robot manipulates an object while this object is in contact with other objects in the environment. These interactions restrict the motion freedom of the manipulated object, and generate contact forces. The task controller must cope with these motion constraints in a safe and user-specified way. FIGURE 1.1 shows the functional components of a general compliant motion task controller, as well as their interaction. The following paragraphs explain the task of each of these functional components in somewhat more detail, and indicate where this text presents new contributions. Remark, however, that the "complete" compliant motion system, as described in this Chapter, does not yet exist, neither for force controlled robots, nor for robots equipped with any other type of sensor(s)!

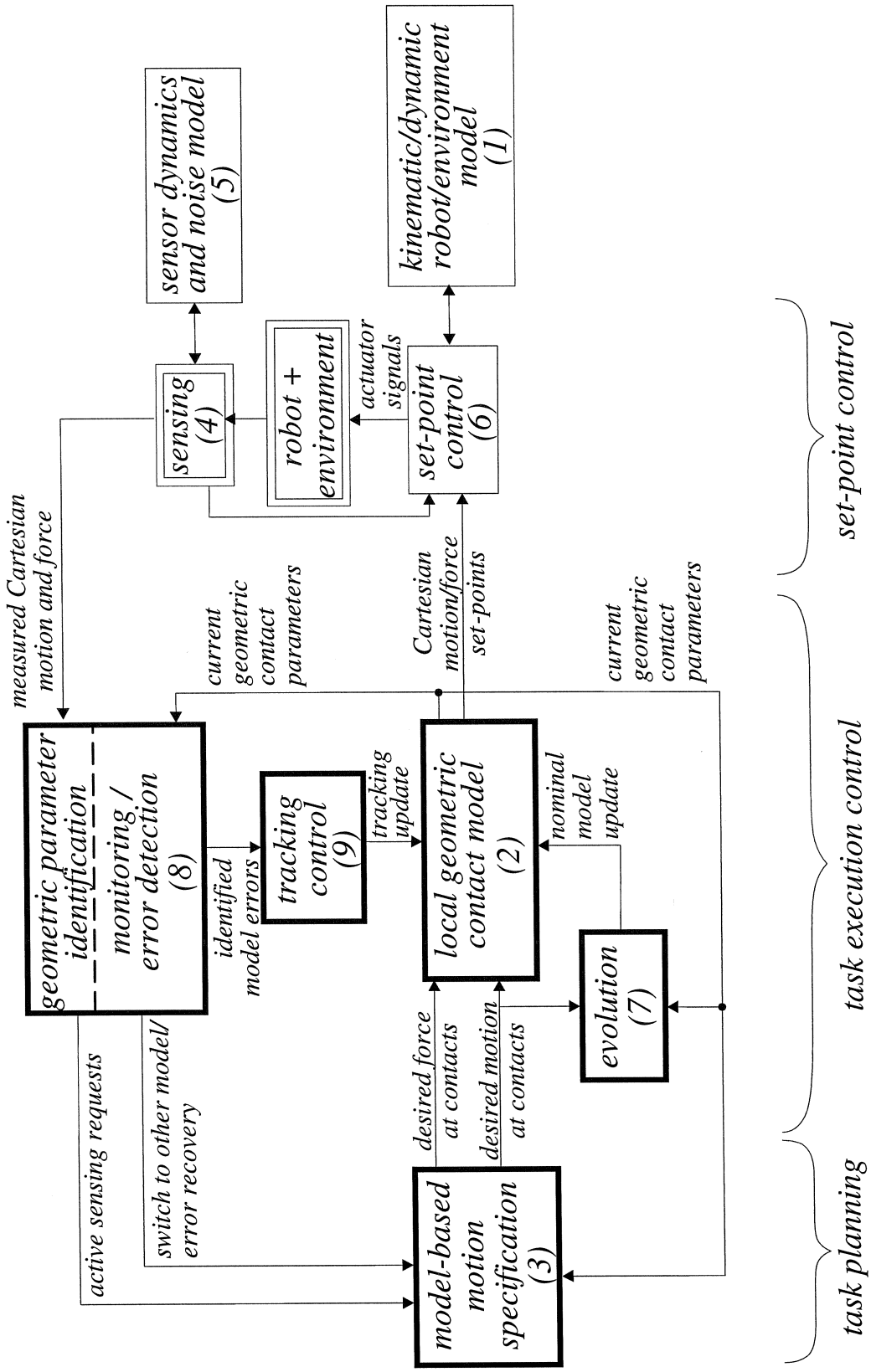


FIGURE 1.1 : *Overview of a general compliant motion control system.* The modules with a bold outline are further elaborated in the following Chapters. The doubly framed modules contain physical components; the other modules consist of software only.

Modelling This is the most important step in the design of a computer controlled system: except for the sensors and the actuators, no components of the system interact with the real physical world, but only with a mathematical abstraction of it. Hence, the quality of this abstract model determines the maximum performance that can be expected from the closed-loop control. The real performance is then determined by how well one succeeds in: 1) filling in the correct model parameters, and 2) implementing a real-time controller for the total model. Good models are a necessary, but not sufficient condition to derive a good motion *feedforward*. Indeed, with a good feedforward specification the task control needs only little feedback and exhibits small tracking errors. A general compliant motion system works with distinct types of models:

1. A *kinematic* and *dynamic model* (module (1) in FIG. 1.1) of the robot arm that moves the *MO*[1] and the *ENV* with which the *MO* is in contact.

2. A *motion constraint model* (module (2) in FIG. 1.1) describing the interaction between the *MO* and the *ENV*: How is the motion freedom of the *MO* instantaneously constrained by the *ENV*, and what reaction forces can occur? A motion constraint model in itself consists of a *geometric* part —Where do *MO* and *ENV* contact each other? What is the *local* surface geometry of both objects at these contact points? What *instantaneous* velocities are allowed for the *MO*?— and a *dynamic* part —How does the contacts influence the *MO*'s dynamic behaviour, as well as that of the robot? The geometric model could also be called the *kinetostatic* model, since the instantaneous and ideal motion freedom and contact forces are completely determined by the objects' geometry at the contact points.

3. A *sensor dynamics and noise model* (module (5) in FIG. 1.1) describes the extra dynamics introduced by the sensor(s), and gives statistical information about how the pure sensor data are corrupted by noise.

This text concentrates on local and instantaneous models of the contact geometry, i.e.:

1. The *type* of the contact is given; e.g., a *vertex-surface* contact.

2. The *geometry* of the objects in the neighbourhood of the contact point is described by *first order* parameters (position of contact point, direction of contact normal) and *second order* parameters, i.e., curvature.

Note that this text is *not* about the planning of contact tasks (so-called *fine motion planning*), which uses the global models of the *MO* and the *ENV* to derive a sequence of contact situations —characterized by the type of the contact, and the part of the *MO* and the *ENV* involved in it— that lead the *MO* to a specified goal position.

[1] "*MO*" indicates the manipulated object, "*ENV*" indicates the environment.

Motion specification The specification module (module (3) in FIG. 1.1) uses the current geometric contact model (module (2) in FIG. 1.1) to generate velocity and acceleration set-points for the robot end effector, as well as a reaction force set-point. These set-points must correspond to a motion of the *MO* that is allowed by the contacts, and to a force that can be generated by the contacts. If the contact model and the kinematic-dynamic model of the robot arm are perfect, a good motion specification module could, in principle, generate motion set-points that make the real robot system behave exactly as desired, even without feedback control. Of course, perfect models do not exist, and feedback control is indispensable. Nevertheless, it remains plausible to compare the "quality" of different motion specification modules. One module is better than the other if: 1) the motion it generates causes smaller tracking errors, with the same feedback controller; 2) it is able to make the *MO* move at higher speeds or under more complicated motion constraints; or 3) it requires simpler motion constraint models and user inputs.

This text presents extensions to the state-of-the-art in each of these quality features.

Sensing (module (4) in FIG. 1.1) The executed motion and the resulting reaction force are measured, at the robot joints and by a wrist force sensor, respectively. In order to interpret these measurements, they must be transformed to the Cartesian reference frames used for specification, control and monitoring of the *MO*'s motion. The corresponding mathematical transformations are straightforwardly derived from the robot's kinematic model. More involved models allow to filter these "pure" Cartesian motion and force data, e.g., to subtract the friction and dynamic forces, to suppress sensor noise, or to detect and quantify the influence —on the transformations of the measurements— of compliance in the mechanical part of the system.

This text assumes *noiseless sensors*, *ideal reaction forces*, as well as a *perfectly stiff* robot arm, objects and sensors.

Control The set-point (or low-level) controller (module (6) in FIG. 1.1) uses feedback algorithms to keep the measured motion and the measured reaction force as close as possible to the reference inputs. Some major keywords in control are: high bandwidth, no overshoot, small tracking errors, stability, robustness and disturbance rejection. To achieve these goals, the set-point controller can —in conjunction with the motion specification module— rely on *dynamic models* of the robot, the sensors, the actuators, the interaction with the environment, etc.

The control component of a compliant motion system has received disproportionately much attention in the robotics literature, which has given rise to countless variations in control schemes. The right-hand side of FIG. 1.1 is nothing more than a conceptual representation of a (Cartesian space) controller module. From the two major compliant motion control approaches —the *hybrid* and *impedance* control formalisms, Raibert & Craig (1981), Hogan (1985), Kazerooni, Houpt & Sheridan (1986), Whitney (1987)— the hybrid approach is most compatible with the modelling, motion specification and geometric uncertainty identification developed in the next Chapters.

This text does not deal with control explicitly, since this is already the most advanced

module in FIG. 1.1, and its current state-of-the-art, even in commercial implementations, has reached a very acceptable and workable level. Hence, major improvements in the performance of compliant motion systems are rather to be expected from other modules, e.g., (automatic) task planning, identification and monitoring, error recovery...

Evolution (module (7) in FIG. 1.1) The motion specification module generates desired velocities and accelerations. These are transformed by the controller into actuator signals for the robot, which move the *MO*. Hence, (some of) the geometric model parameters change, and the model needs to be updated. Part of this update is derived from the actual model itself: this *nominal evolution* corresponds to the change in the motion constraint generated by the motion of the *MO* in the case of a perfect model. The nominal evolution predicted by different models differ, and again a quality classification is possible: one model is better than the other if it predicts the nominal evolution of the motion constraint more accurately. A good nominal evolution is important because it facilitates the correct interpretation of the sensor measurements: since the *MO* is in continuous motion, the parameters of the contact geometry change continuously too. In general, this evolution cannot always be measured directly or completely, and hence it must be (partly) estimated.

This text does not use dynamic models. This means that it assumes the executed velocity to be equal to the desired velocity. This assumption is motivated by the facts that: 1) for many commercial robot arms, the internal position or velocity control loops have high bandwidths, especially for "slow" tasks such as compliant motion, and 2) the real motion of the robot can always be *measured*, even if specified and real motion differ. Hence, the nominal evolution is simply the integral of the (measured or specified) velocities and accelerations.

Identification and Monitoring (module (8) in FIG. 1.1) The measurements not only serve as inputs to the set-point controller, but also to estimate inaccurately known parameters in the models. Some of these parameters cannot be measured directly, and must be observed based on the current state of the model itself. This observation could need more data than generated by the nominally specified motion. In this case the identification module superimposes some "active sensing" actions upon the nominal motion. A good uncertainty identification module increases the task execution performance in poorly known environments.

During task execution, the identified updates are constantly *monitored*, in order to decide whether or not the current type of motion constraint is still appropriate. A sudden transition in the contact situation would be signaled by abrupt changes in the measurements or the estimator results. It is the monitor's responsibility to detect such transitions. If the detected transition complies with the nominal task sequence, it switches to the next local contact model and the corresponding motion specification. If the detected transition cannot be explained by the nominal plan, an error recovery module should be consulted. However, error recovery is currently among the least developed modules in the total compliant motion system.

Model-based identification of uncertainties in the *geometric* motion constraint model

are the most original contributions of this text.

Tracking (module (9) in FIG. 1.1) The outputs from the identification module are used to adapt the model. This activity is called *tracking of the model uncertainties.* It is very similar to the set-point control module: the latter wants the mechanical system to follow a desired trajectory with minimum tracking errors; the former wants the model to follow the real world with minimum tracking errors. The combination of 1) model-based motion specification, 2) feedforward and feedback set-point control, and 3) uncertainty tracking, is often referred to as *adaptive control.*

It would in general not be a good idea to instantaneously compensate the identified errors: a change of the model (indirectly) changes the specified motions; these changes excite the dynamics of the total system, which could result in unstable behaviour. Hence, implementing a tracking module requires some feedback control loop design. Tracking generally achieves lower bandwidths than the robot set-point control, because it is implemented as an external loop.

This text pays only superficial attention to the design of a tracking controller; the experiments described in Chapter 7 perform reasonably well with only simple proportional tracking feedback.

1.2 Compliant motion tasks: the Task Frame Formalism

This section lists some compliant motion tasks that the current state-of-the-art can specify. It indicates the common features of all these tasks, as well as how much of the modules in the general compliant motion system of FIG. 1.1 is required to execute them.

The compliant motion tasks of the kind presented below belong to the paradigm of the *Task Frame Formalism —TFF* for short— also called the *Compliance Frame Formalism.* This paradigm works with the assumption that, for every motion constraint, an orthogonal reference frame exists —the so-called *task frame* or *compliance frame*— in which the force controlled and velocity controlled directions are orthogonal and coincide with the frame's axes.

The most commonly described and implemented tasks —Gerelle(1978), Van Brussel & Simons (1979), Mason (1981), Hanafusa et al. (1981), De Schutter & Van Brussel (1988), Whitney (1987), Bennett & Hollerbach (1989), Yoshikawa & Sudou (1993)— are:[2]

1. *Opening a door* (FIG. 1.2), *turning a crank* (FIG. 1.3), or *screwing a bolt* (FIG. 1.4). These systems are *mechanisms*, with only one degree-of-freedom. The compliant motion controller makes the programmed motion correspond toa physically allowed motion by adapting its set-points in such a way that the generated reaction forces remain within safe limits.

[2]The task frame is drawn in each of the FIGURES 1.2-1.11.

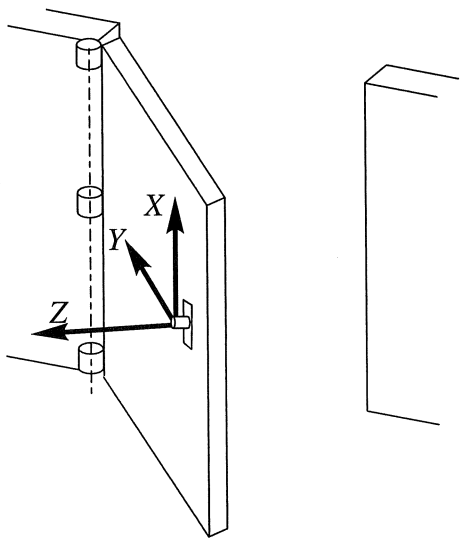


FIGURE 1.2 : *Opening a door.*

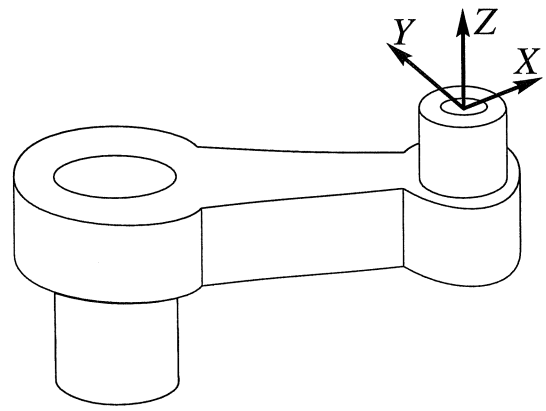


FIGURE 1.3 : *Turning a crank.*

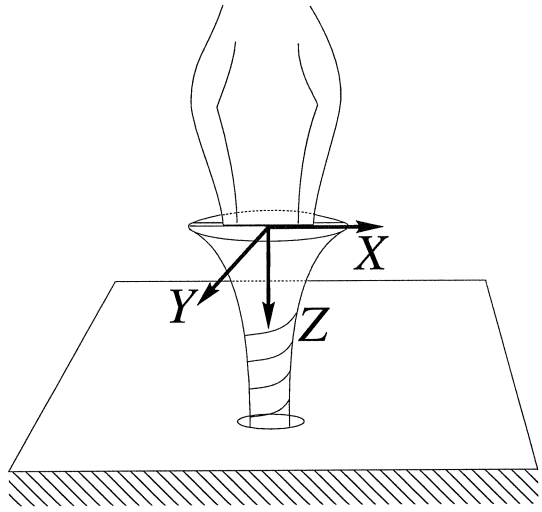


FIGURE 1.4 : *Turning a screw.*

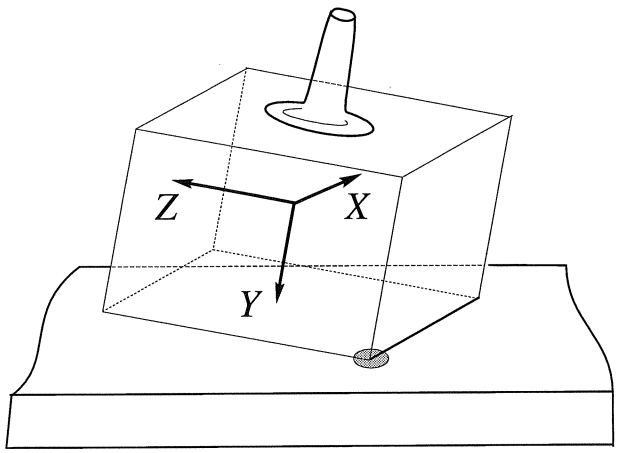


FIGURE 1.5 : *Aligning a block*

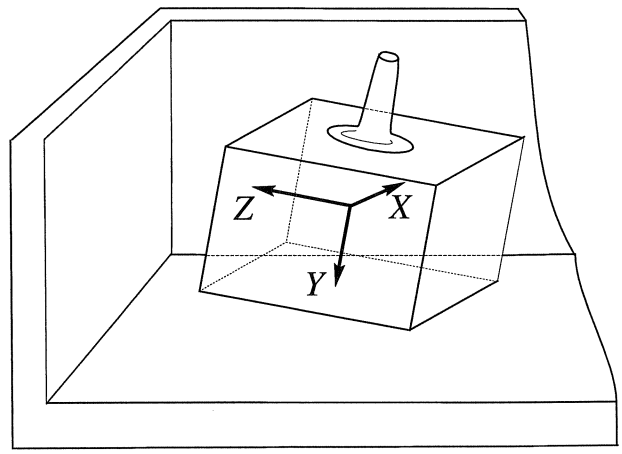


FIGURE 1.6 : *Placing a block in a corner.*

2. *Aligning a block with a surface*, FIG. 1.5. In this task, force control serves two purposes: 1) it pushes the block against the surface with a certain force in order to be sure that there is indeed a contact; 2) it regulates all reaction torques to zero, in order to be sure that the contacting face of the block is aligned with the surface.

3. *Placing a block in a corner*, FIG. 1.6. This is equivalent to three of the previous tasks in parallel, i.e., one for each face of the block that must be brought into contact with the environment.

4. *Sliding a block over a surface*, FIG. 1.7. This task is only slightly more complicated than aligning a block with a surface: now also the desired tangential velocity (magnitude and direction) has to be specified and controlled.

   In all of the above-mentioned tasks the controller makes the robot "comply" to the motion constraint. (This behaviour has given its name to the whole field of constrained motion under active force control.) This kind of tasks needs no explicit tracking module: controlling the desired force and velocity set-points suffices. This makes the execution robust against uncertainties in the shape and relative positions of the objects involved, and against poorly tuned control gains.

5. *Inserting a round peg*, FIG. 1.8. This task is only slightly more complicated to specify and execute than the previous tasks: specify an insertion velocity along the direction of the hole's axis, an angular velocity about the hole's axis, and control all forces and torques to zero. However, this task is much less robust, since wedging and jamming can occur if: 1) the directions of the peg's axis and the hole's axis are not well aligned at the start of the task, or 2) the control gains for force and torque components are not well tuned.

6. *Sliding an edge over a surface*, FIG. 1.9. Compared to sliding a block over a surface,

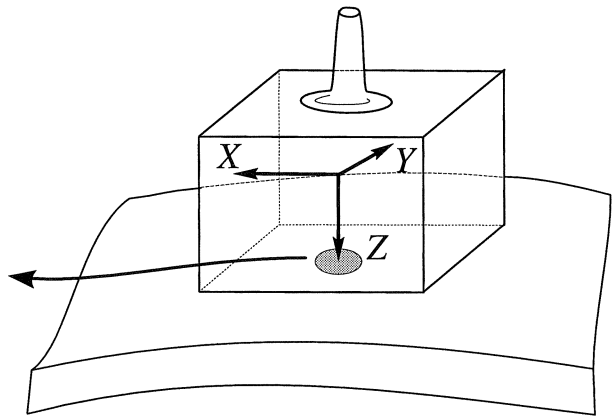


FIGURE 1.7 : *Sliding a block over a surface.*

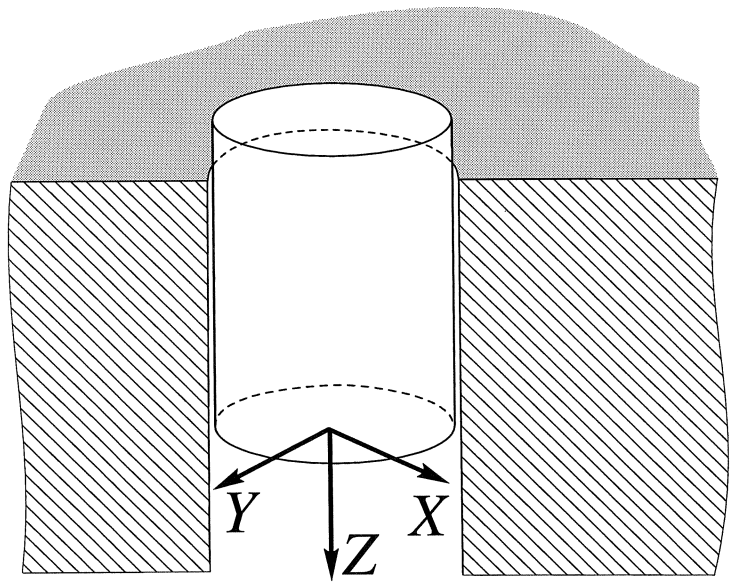


FIGURE 1.8 : *Inserting a round peg.*

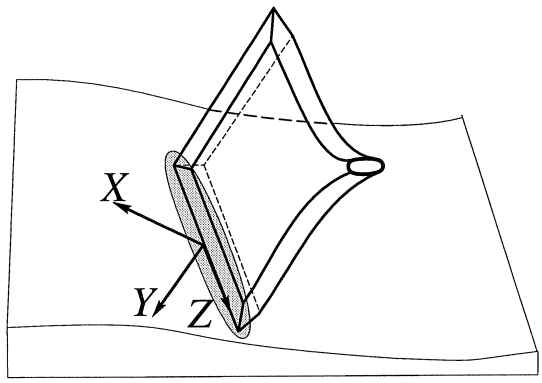


FIGURE 1.9 : *Edge sliding.*

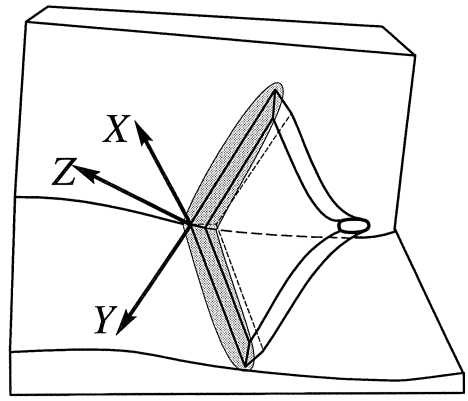


FIGURE 1.10 : *Following a 3D seam.*

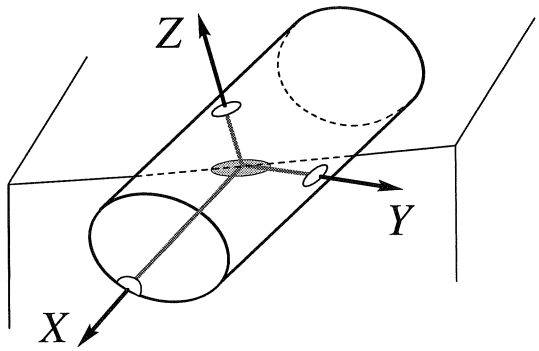


FIGURE 1.11 : *Sliding an edge over an edge.*

this task has one *tracking degree-of-freedom*: rotation of the *MO* about the edge. This degree-of-freedom needs a tracking module, because the angle between a reference frame on the *MO* and the contact normal cannot be measured directly.

7. *Following a 3D seam*, FIG. 1.10, with a *MO* that fits into this seam, i.e., it has two edge contacts with the seam. This task is equivalent to two of the previous tasks in parallel, hence it needs tracking for the angular degrees-of-freedom of the *MO* about the two edge contacts. The task is also called *compatible seam tracking*, in contrast to the *incompatible seam tracking* task of FIG. 1.13.

8. *Sliding an edge over an edge*, FIG. 1.11. Compared to sliding an edge over a surface, this task introduces one more degree-of-freedom —the rotation of the *MO* edge about the *ENV* edge— and one more tracking parameter —the position of the contact point on the edge of the *MO*.

Since it is not strictly necessary for the execution of the tasks, the implementations reported on in the literature —with the notable exceptions of surface and contour following— seldom try to extract explicitly all information about the instantaneous contact geometry contained in the sensor signals. For example, in the opening-a-door task, it is not that

difficult to estimate the position and orientation of the door's hinge, De Schutter (1988). However, this knowledge can increase the quality of the task execution (in terms of execution speed, force errors, error detection and recovery) but is not really necessary to make the task succeed. Moreover, most implementations artificially restrict the capabilities of their compliant motion system as follows:

- The position of the task frame with respect to the robot end effector or the environment is either constant, or easily updated by some sort of evolution module (e.g., translation along the X axis in the *edge-edge* sliding task, FIG. 1.11) or by the tracking module (e.g., *3D seam following*).

- The full motion freedom of the *MO* with respect to the constraint is not fully exploited, or sometimes not even modelled. For example:

  1. Placing a block in a corner: in this task, one, two or even three contacts can occur between the *MO* and the corner. These intermediate contact situations are not explicitly modelled. One of the consequences is that the result of the task specification cannot be predicted if the initial position of the *MO* is far from the desired goal position: trying to control away the measured torque components can then indeed result in the *MO* rotating in the wrong direction. On the other hand, it is possible —e.g., see Chapter 7— to find a motion specification in each of the intermediate contact situations that brings the *MO* closer to its desired final position, regardless of the uncertainty in the current relative position between block and corner.

  2. The tracking degrees-of-freedom, if available, are most often used to keep the orientation of the *MO* constant with respect to the contact normal. Hence, some available motion degrees-of-freedom are not explicitly specified, but filled in online by tracking. This is illustrated by the example of *2D contour following*, FIG. 1.12 (Hanafusa et al. (1981), Whitney (1987), De Schutter & Van Brussel (1988)): the motion constraint model for this task is that of the *edge-surface* contact, but with the *MO* constrained to move in a plane. A force has to be exerted along the direction of the contact normal, and two degrees-of-freedom remain: a tangential velocity along the contour, and an angular velocity about the edge. This angular degree-of-freedom is filled in by tracking.

  3. Compatible 3D seam following, i.e., the *MO* has the same shape as the seam's cross section, but it exactly fits into the seam at only one relative position, i.e., when the plane through the edges of the *MO* is orthogonal to the seam. In this position, only one degree-of-freedom remains. Allowing the *MO* to follow the seam with a different contact configuration —for example, two *vertex-surface* contacts, FIG. 1.13— increases the available degrees-of-freedom for the manipulating robot. These extra degrees-of-freedom can be specified by the user, or by an automatic redundancy resolution or task optimization algorithm.

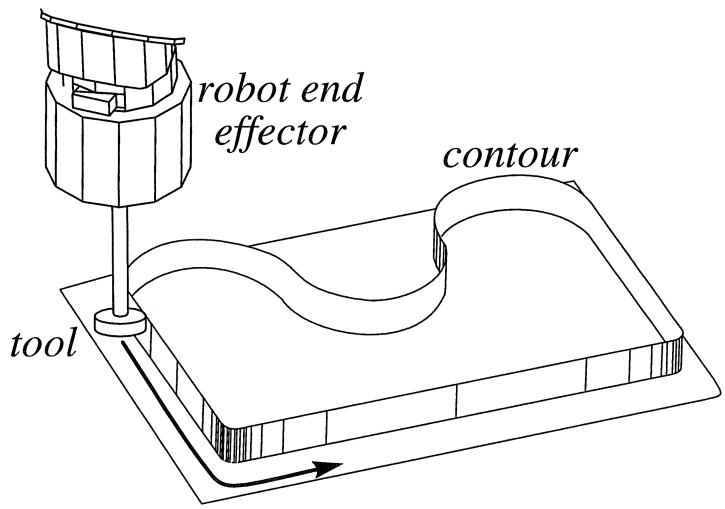


FIGURE 1.12 : *2D contour following.* This is a special case of the *edge-surface* task, in which the *MO*'s motion is constrained to a plane.

In summary, the task frame is mostly used as a *time invariant* geometric motion constraint model. This indeed simplifies the integration of the different modules in the compliant motion system described in Section 1.1: 1) neither the motion specification, nor the contact type change during the task, 2) sensing reduces to a simple and fixed selection of components from the measurements (the famous "selection matrices" in the hybrid control approach!), 3) tracking adapts the position or orientation of the model with respect to the robot and is limited to simple and intuitive relationships between a restricted subset of the measurement components (Yoshikawa & Sudou (1993), De Schutter & Van Brussel (1988)), and 4) this approach is fully compatible with the major tendency in robot force control, i.e., hybrid control, Raibert & Craig (1981), Whitney (1987), De Schutter & Van Brussel (1988).

1.3 The Contact Frame Formalism

The main motivation of this text is to model, specify and identify *arbitrary* motion constraints. This means that: 1) *MO* and *ENV* have arbitrarily complex shapes; 2) multiple contacts may occur in parallel; 3) the exact shape and position of the contacting surfaces must not be completely known a priori; and 4) the full motion freedom allowed by the motion constraint is modelled and must be available for the task specification. The surfaces of the objects are assumed to be *smooth*, in a mathematical sense —i.e., continuous first and second derivatives— as well as in a physical sense —i.e., good surface finishing, such that friction is not excluded but occurs only with a rather low friction coefficient. Arbitrary motion constraints are, by definition, *time varying*. So it is impossible to define a unique frame with the same simplifying features as for the tasks in the previous Section. Some

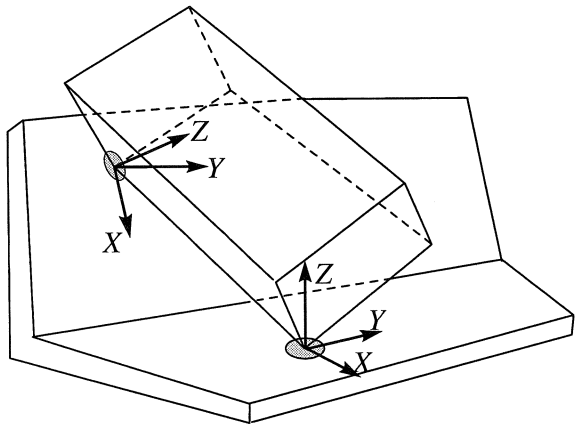


FIGURE 1.13 : *Incompatible seam following.*

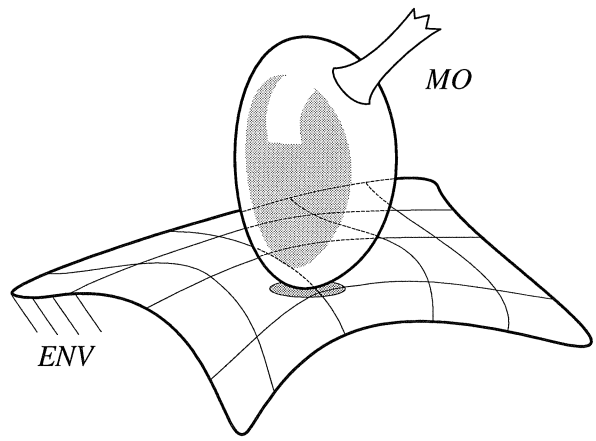


FIGURE 1.14 : *Surface-surface contact.*

examples of time varying compliant motion tasks are:

1. *Incompatible seam following*, FIG. 1.13. The task consists of following a 3D seam —i.e., two intersecting smooth surfaces— with a tool that does not fit nicely into the seam angle. The time variance results from the continuous change of the relative position and orientation of the two point contacts, with respect to each other as well as with respect to the robot end effector.

2. *Surface-surface contact*, FIG. 1.14. This is the most general contact type. Its time variance results from the continuous change of the position of the contact point and the orientation of the contact normal. These changes are much more complicated to track than in the example tasks of Section 1.2.

3. *Peg-on-hole*, FIG. 4.6. The contact situation between a round peg on a round hole is time varying, since moving the peg on the hole changes the three contact points and contact normals.

The following Chapters of this text present a *unifying theoretical framework* to integrate the following system modules for arbitrary motion constraints:

Geometric motion constraint modelling. The geometry of the contact is described by its local *first* and *second order* parameters: the position of the contact point and the direction of the contact normal are the first order parameters; the curvature represents the second order geometry.

Motion specification. Also to first and second order, i.e., ideal reaction forces and instantaneous velocities (*first order*), and accelerations (*second order*).

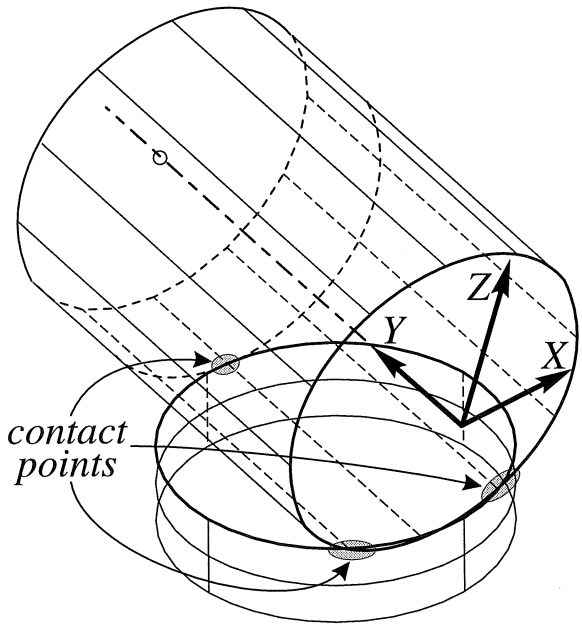


FIGURE 1.15 : *Peg on hole.*

Identification of the uncertainties in the first and second order geometric parameters. With a slight abuse of terminology, these uncertainties are called *first* and *second order uncertainties* too.

For every possible contact situation, this theoretical framework delivers information about: 1) the nominally allowed motion degrees-of-freedom, 2) the possible ideal reaction forces, 3) a model of the geometric uncertainties with a minimum number of parameters, 4) sets of linear identification equations linking the uncertainties to the force and motion measurements, and 5) an analysis of the identifiability of the uncertainties. To the best of the author's knowledge, the robotics literature has not yet reported on any comparably integrated and general effort in the area of modelling, motion specification and uncertainty identification. The presented research has been marked by the following design guidelines:

Simplicity. The models are simple and intuitive, with as few parameters as possible. Examples of often practically justifiable simplifications are: both *MO* and *ENV* are *rigid bodies*, and non-ideal effects —such as *friction*, *compliance*, *surface roughness* and *texture*— are not included in the models, but regarded as disturbances.

Flexibility. The theory applies to all possible motion constraints between rigid bodies, and describes the contact interaction physics at a sufficient level to allow reliable execution with real robot arms.

Adaptability. The models are designed to be adaptable, by feedback from an uncertainty identification module.

Efficiency. The uncertainty identification algorithms are developed for on-line, recursive use.

Compatibility. All TFF tasks are special cases of the presented approach. However, viewing them in the new light of this approach has led to original extensions of the TFF, such as new identification equations.

Invariance. The non-invariance of the theoretical foundations has plagued the literature on hybrid force control since its inception: in most references, the control or sensor processing approaches depend on the reference frames in which the motion and forces are expressed, as well as on the physical units used, Lipkin & Duffy (1988), Duffy (1990). The utmost care has been taken in this text in order that this non-invariance problem does not show up.

1.4 Overview of the text

The major innovative ideas of this thesis are: 1) the consistent use of the *kinematics of mechanisms* as the unifying framework for the integration of modelling, specification and identification of compliant motion tasks; 2) the use of the *first and second order* geometric parameters of the surfaces of *MO* and *ENV* as basic model parameters; and 3) the derivation of *identification equations* for uncertainties in these parameters, based on the general physical properties of consistency and reciprocity. These ideas fit into the general compliant motion system of Section 1.1 as follows:

1. Contacts are modelled by *virtual manipulators*, linking the *MO* to the *ENV*, and subjecting it to the same motion constraint as produced by the physical contact.

2. The model for these virtual contact manipulators is based on the first and second order geometric parameters of the contact surfaces. The mathematics are most easily represented with respect to a "natural" reference frame —the so-called *contact frame*— at the contact point; this natural frame is deduced from the geometry of both contacting objects in the neighbourhood of the contact point.

3. Uncertainties are modelled by *uncertainty joints* in the virtual contact manipulators.

4. The total motion constraint, consisting in general of several parallel contacts on the *MO*, is mathematically described by the *loop closure equations* of the complex kinematic chain formed by the serial concatenation of, on the one hand, the real manipulator, and, on the other hand, all parallel virtual contact manipulators between *MO* and *ENV*.

5. Analysis of the loop closure equations reveals that only n_f of the joints in the complex kinematic chain can be *driven* independently (n_f is the number of degrees-of-freedom of the *MO* in the current contact situation). This means that the motion of all other

joints, as well as of the *MO*, is unambiguously determined by the joint velocities of these n_f driving joints. *Motion specification* for the *MO* then boils down to choosing appropriate driving joint velocities.

6. Given the kinematic structure of the total constraint, it is straightforward to transform the sensor measurements to the reference frames in which they must be interpreted. All relevant reference frames, such as the contact frame, are fixed to one of the links of the kinematic chain. Hence, their *evolution* follows directly from the known (internal) motion of the kinematic chain.

7. *Identification* means closing the loops in the chain when the loop closure is disturbed by the geometric uncertainties in the model. An a priori analysis of the identifiability of a given compliant motion task can be derived from a kinematic mobility analysis of the total kinematic chain.

The next Chapter reviews the literature related to the material presented in this text. It turns out that very little research exists on the modelling, motion specification and geometric uncertainty identification aspects of compliant motion tasks.

Chapter 3 elaborates on the technical details of how to represent motion (*twists*) and forces (*wrenches*), how to model the instantaneous motion freedom of a general manipulator by means of its *Jacobian matrices*, and how to calculate the derivative of these Jacobians with respect to time and with respect to one of the joints in the manipulator. This material may seem rather heavy to digest, but it is extensively used in the other Chapters. Moreover, it contains important original material about on the one hand the different mathematical representations of twists and wrenches, and on the other hand the calculation of a Jacobian's derivatives.

Chapter 4 details the construction of motion constraint models, the so-called "virtual contact manipulators," based on the first and second order contact geometry. It also describes their mathematical representation in terms of *twist* and *wrench Jacobians*.

Chapter 5 incorporates the description of motion constraint uncertainties in the nominal virtual contact manipulators. It derives the dependencies of the twist and wrench Jacobians on these uncertainties. If multiple contacts occur between the *MO* and the *ENV*, it is sometimes possible to reduce the number of uncertainty parameters; this Chapter presents one possible systematic uncertainty reduction procedure.

Chapter 6 describes how to derive identification equations for the motion constraint uncertainties, starting from the physical principles of consistency and reciprocity. It shows that both approaches give equivalent results. These are illustrated by the example of 2D contour following. This Chapter also looks deeper into the practical problems of *identifiability* of uncertainties, and the influence of *friction* and *structural compliance*.

Chapter 7 presents some real world experiments illustrating the theoretical results, as well as other potential applications.

Chapter 8 summarizes the major contributions of this work, tries to critically examine their relevance and potential for robot compliant motion, and outlines further research to solve some of the remaining (and newly introduced!) problems.

Finally, the Appendices give more mathematical details on subjects in the other Chapters that are not necessary for a good understanding of the text.

1.5 Conclusions

This thesis work is about compliant motion with time varying motion constraints. It has been motivated by:

- Some shortcomings in the current state-of-the-art —the Task Frame Formalism— for motion specification and uncertainty identification: the inability to cope with time varying motion constraints, limited and ad hoc identification and tracking algorithms, the lack of generality, and the abundant usage of non-invariant approaches.
- The desire to strictly separate control from modelling, specification and identification, without loosing sight of the need for integration of sensing, planning and control.
- An emphasis on simplicity and real-time implementation.

The theory of the kinematics of mechanisms is consistently used as the mathematical framework by which the contributions of this work are integrated: geometric motion constraint modelling for arbitrary motion constraints, first and second order motion specification (i.e., ideal reaction forces, and velocities and accelerations), and identification of first and second order geometric uncertainties (i.e, the position of the contact points, the direction of the contact normals, plus curvature information).

The framework developed in this thesis was given the name of *Contact Frame Formalism* —or *CFF* for short— to stress the change of viewpoint in looking at the compliant motion problem: the TFF originated in the context of hybrid force control, which resulted in a *robot-oriented* and *control-oriented* approach; this thesis adopts a more *object-oriented*, or *model-oriented*, approach, modelling first each of the contacts between *MO* and *ENV* independently from any set-point control module, and with respect to the local *contact frame*. Only then the interactions between the different contacts are systematically taken into account, and their influence on the robot's motion freedom are calculated.

Finally, a major practical advantage of the kinematic approach followed in this thesis, is that it allows to use much well-established terminology, and many tools and algorithms: twists and wrenches (i.e., rigid body motion and reaction force), Jacobian matrices and transformations, loop closure solving, redundancy resolution techniques, ...

Chapter 2

Literature survey

2.1 Introduction

This Chapter discusses historically relevant and recent state-of-the-art contributions in the robotics literature that have connections to or impact on the work described here. This survey does not cover all modules of the general compliant motion system of FIG. 1.1, but concentrates on the key subjects treated in this text: geometric modelling of motion constraints (Section 2.3); application of these models to motion specification (Section 2.4); modelling of geometric uncertainties (Section 2.5); and identification of these uncertainties (Section 2.6). Firstly, however, the next Section briefly discusses the *non-invariance* problem that plagues the hybrid force control literature already since its inception.

2.2 Non-invariance of the Task Frame Formalism

In the context of constraint modelling and motion specification for robot compliant motion, Mason (1981) is the seminal paper. Its importance, however, is limited to this historical fact: Mason was the first to describe the Task Frame Formalism (TFF) in the literature, but only on a very intuitive level, without consistent theoretical foundation. On the one hand, the TFF presents a simple approach to model motion constraints that resulted in successful implementations —done by others— but on the other hand, it inseminated the research area of force controlled compliant motion with a *non-invariant* "theory."

Non-invariance means that theoretically derived results depend on the physical units in which the results are expressed, as well as on the reference frame chosen to represent the results. This non-invariance is due to the fact that the intuitively appealing concept of "orthogonal" force controlled and velocity controlled directions is physically inconsistent: one applies the Euclidean norm to forces and velocities, but no such norm exists for rigid body motions and reaction forces! This fact is clearly mentioned or proved several times during the last decade (e.g., Karger & Novák (1985), Lončarić (1987), and Park (1994)), but seems to be ignored by the majority of the compliant motion research community. Using the Euclidean norm to derive features of a compliant motion system can lead to incorrect

results. Probably the most prominent example of such incorrect results are the "selection matrices" that show up in many papers on hybrid force/position control: the measured forces and velocities are "projected" onto the force controlled and position controlled directions of the motion constraint by means of a diagonal matrix containing only ones and zeros. The selection matrices for forces and velocities are complementary in the sense that if $S_v = \text{diag}(0\ 1\ 0\ 1\ 1\ 0)$ is the velocity selection matrix, then $S_f = \text{diag}(1\ 0\ 1\ 0\ 0\ 1)$ is the force selection matrix. However, these selection matrices should only be used in very particular reference frames (the "task frames" of Chapter 1) but nevertheless they appear in many "general" force control schemes.

In 1988, Lipkin & Duffy (1988) published a first article describing this fundamental and frequently occurring non-invariance problem. Maybe because the contents of their paper, as well as the vocabulary they used, were too far off the mainstream in the compliant motion research field, the message was not understood. Duffy (1990) made a second attempt, with simpler argumentation, but published (or, maybe, *could* only publish) it in a minor robotics journal. Hence, the majority of the researchers in the compliant motion field are still not aware of the problem, or do not fully understand its implications. Griffis & Duffy (1991) published a simple invariant force control law, but few researchers seem to have adopted it.

2.3 Geometric modelling of motion constraints

Most papers deal only with the first order geometry of the motion constraints, i.e., the position of the contact point, and the direction of the contact normal.

Kobayashi (1985) models multiple fingertip contacts, to describe grasping with robotic hands. He introduces "passive joints," which this text generalizes to "virtual contact manipulators."

West & Asada (1985) use a similar kinematic contact model —*virtual arms*— to model the nominal motion freedom of a "braced" robot arm: the arm makes contact with a support plane, at some small distance from its end effector, in order to increase the mechanical stability of the manipulation. Of course, bracing reduces the nominal motion freedom of the manipulator. The paper describes how to take the influence of the bracing contact into account in the design of a hybrid force/position controller.

Hirai & Asada (1993) apply the theory of *convex polyhedral cones* to several research areas in robotics. To model compliant motion applications, they describe the *Admissible Displacement Set*, which is a first order model of the motion constraint (only displacements, no forces). The main contribution of this approach is that it takes the unidirectionality of contacts into account explicitly; unidirectionality means that one can only "push" at a contact, and not "pull."

Models and motion specifications that make full use of the second order geometry of the contacting objects (i.e., their curvatures in the neighbourhood of the contact point) appear primarily in research papers on robotic hands. However, in this context the main

sensory input is assumed to come from tactile sensors on the fingertips, i.e., at the contact points themselves. This avoids the problem of having to *observe* the contact parameters, since they are now *sensed* more or less directly. The research area of robotic hands restricts itself to manipulation of an object in free space. For a manipulated object in contact with the environment, it is unrealistic to assume the presence of tactile arrays at each contact point. For assembly tasks this is even totally impossible.

A second assumption that is only plausible in the context of grasping, is that friction acts as a full motion constraint, i.e., rolling is possible, but slipping undesirable. For the applications aimed at in this text, friction is an undesirable disturbance and both rolling and slipping are possible.

Montana (1988) is a standard reference for second order motion constraint modelling: it introduces the classical differential geometry theory in the area of constrained manipulation. Montana presents transformations from pure differential geometric concepts, such as the curvature form and a metric, into motion parameters of the manipulated object, such as allowed velocity and acceleration. The material applies to grasping with smooth fingertip contacts only. The models are described in Cartesian space, and parameterized by u, v coordinate patches. These kinds of models are not very appropriate for on-line use and adaptation. Hence, this reference, as well as most of its followers, present no experimental implementations. Notable exceptions are Sarkar, Yun & Kumar (1993) and Paljug, Yun & Kumar (1994): these references apply Montana's contact model and dynamic manipulator models to roll a sphere between two manipulators.

Rimon & Burdick (1993, 1994) make a very good plea for using second order geometric models. They state that the "source of deficiency of first order theories is that relative mobility is *not an infinitesimal notion, but a local one*" (the italics are theirs). While their objectives are only in grasping and design of fixtures, this statement applies without change to compliant motion too.

Motion constraint modelling in this thesis makes extensive use of kinematic chains. Angeles (1988, 1993) and Orin & Schrader (1984) are the main references for the definition and representation of kinematic chains and their Jacobian matrices. However, the concept of the time derivative of a manipulator's Jacobian matrix, as well as the derivative with respect to a joint angle, are insufficiently treated in the literature. Hence, Chapter 3 is dedicated to this material.

2.4 Motion specification

Motion specification for constrained systems is always closely linked to how the constraints are modelled: while the model describes the *allowed* motions, motion specification *chooses* one possible motion from this set, in function of the task to be performed. For second order models, the motion specification is, inevitably, only *instantaneous*, i.e., in terms of velocities and/or accelerations. *Displacement* type of motion specification is possible in

first order (i.e., polyhedral) models only. Hence, automatic compliant motion planners — more commonly known as *fine motion* planners in this particular research area – use polyhedral models, and only straight line motions or pure rotations. Laugier (1989) and Xiao (1992) are representative examples.

The grasping papers referred to in the previous Section all describe in more or less detail how their second order models can be used to specify a motion allowed by the constraint. Montana (1988) derives *contact equations*, that describe the relative motion of two smooth and rigid surfaces over each other. This is exactly the type of motion considered in this thesis too. The fundamental differences between both works are: 1) Montana assumes complete knowledge of the surfaces; 2) he assumes the availability of tactile arrays at the contact point; 3) motions are specified for only one single contact; 4) only point contacts are analyzed; and 5) no experiments are reported.

Kobayashi (1985) gives nominal motion specification for rolling only.

Roth and co-workers have published many papers on nominal motion specification for surfaces moving over each other without loosing contact. Ohwovoriole & Roth (1981) defined the important first order motion properties of *reciprocal*, *contrary* and *repellent twists*, i.e., motions of the manipulated object that, respectively, maintain, violate, and break the contact. Kerr & Roth (1986) derive *differential equations* describing *rolling* of two smooth surfaces over each other. The paper is mainly concentrated on other aspects of grasping; no applicable solutions to these general equations are given, and no experiments are reported. Cai & Roth (1986, 1987, 1988) and Cai (1990) give a thorough and complete description of the nominal kinematics for the relative motion of objects under point and line contacts. First and second order motion specifications are derived, and for some cases even higher orders are included. The first, second and third order parameters of curves and surfaces are the geometric basis for the derivations. The motions are only nominal, i.e., no effort is done in deriving on-line identification algorithms. No experiments are reported, and tactile sensing at the contact points is explicitly assumed. The common kinematics reference for all these papers by Roth and co-workers is Bottema & Roth (1979).

Reynaerts & Van Brussel (1993b) present a kinematic model —a *slider-crank* mechanism— to model and specify the motion of a curved object, rolled between two fingers. The rolling is performed using the *full envelope* of the fingers, instead of only the fingertips.

Rimon & Burdick (1993, 1994) use their second order motion constraint model to define *second order roll-slide motions*. These are derived from the physical principle of *reciprocity* —this is the terminology used in this thesis, not in the cited paper— that is also the foundation for the identification theory presented in Chapter 6.

The above-mentioned references are not directly applicable to compliant motion, for reasons already mentioned before. Some references that do take into account real-time and real world applications come from set-point control research in compliant motion. Whitney (1987) is one of the most accessible introductions to the set-point control aspect of compliant motion. It also superficially discusses motion specification and tracking for 2D contour following ("edge following") with stiffness, damping and hybrid control.

West & Asada (1985) use the theory of closed kinematic chains to calculate the motion freedom remaining after bracing, and apply pseudo-inverses to construct projection operators on the force and position controlled subspaces. These pseudo-inverses, however, use Euclidean norms, which give rise to non-invariant results. No experiments are reported.

De Schutter & Van Brussel (1988a) introduce a language to specify compliant motions in the Task Frame Formalism. The language is able to specify on-line generated tracking velocities. Van de Poel & al. (1993) describe an extended implementation of this language. One of these extensions —the possibility to add arbitrary user-defined feedforward to the nominal motion specification— has been used in this thesis to perform all experiments of Chapter 7. De Schutter & Van Brussel (1988b) describes the set-point controller underlying this implementation. Demey & De Schutter (1994) present model based motion specification for 2D contour following, with the models coming from different sources: CAD, vision, or a previous task execution. These sources give second order information, directly in the form of tracking feedforward: an angular velocity of the task frame proportional to the curvature of the contour. The result is an order-of-magnitude increase in the task execution speed.

2.5 Modelling of geometric uncertainties

Modelling of uncertainties is a basic activity in the context of assembly planning, Laugier (1989), Xiao & Volz (1988), Desai & Volz (1989), Xiao (1992). The feasibility of a planned task, or its guaranteed successful completion, are derived from *stochastic* models of the uncertainties: some kind of "uncertainty ball" for position uncertainty, and "uncertainty cone" for orientation uncertainty. These stochastic models are not applicable to deterministic on-line control. However, this research does have an important complementarity to the material of this thesis: when an ongoing compliant motion reaches a transition in the contact situation —this is signaled by the monitoring module, FIG. 1.1— a next action has to be scheduled. This sequence of actions could be generated by a planner. If there are (large) uncertainties in the system, it is well possible that the encountered contact situation differs from the nominally expected one; in this case an on-line (re)planning module is necessary.

De Schutter & Van Brussel (1988a) describe *orientation* uncertainties of the task frame: these uncertainties model the inaccurate knowledge about the direction of the contact normal. The position of the contact point must be known.

The context of manipulator calibration presents similar ideas to model uncertainties as applied here, in the models of the virtual contact manipulators. Bennett & Hollerbach (1989) use uncertainties in the nominal Denavit-Hartenberg parameters describing the closed kinematic chain formed by the robot and its (bidirectional) contact with the environment. Mirman & Gupta (1993) describe a robot calibration strategy that runs along the same lines as what this thesis proposes for general motion constraints: it perturbes the nominal Jacobian matrix of a manipulator by the kinematic errors (i.e., "uncertainties")

to be identified. Then, it calculates the influence of these errors on the motion freedom of the manipulator. This is equivalent to the derivatives of the Jacobian with respect to these uncertainties, as described in Chapter 5. However, this reference presents its contributions in an ad hoc and hence not very general way.

2.6 Identification of uncertainties

Most papers restrict the sensing and identification problems to the use of the measured force only, possibly after some sort of deterministic or stochastic filtering, and to the estimation of the position of the contact, as well as the direction of the contact normal. This text, on the contrary, presents both force based and motion based identification techniques.

It took a long time before the idea and use of continuous identification and tracking really penetrated in compliant motion research. People have long looked (and some still do!) at tracking as an AI activity: "Do this action, then sense, then reason about the result, then schedule the next action, etc." For example, Ando & Miyake (1981), Lu, Duffie & Bollinger (1982), Ahmad & Lee (1990). In fact, tracking can be implemented as an on-line control activity, Asada & Hanafusa (1980), Hanafusa & al. (1981), Wampler (1984), De Schutter & Van Brussel (1988a).

2.6.1 Force based identification

Identification based on information generated by the force sensor can roughly be classified in three groups:

1. The identification of uncertainties in the geometric parameters of a given contact type. (This is the subject of this thesis.)

2. The identification of the type of the contact. (Not considered in this text.)

3. Some sort of hybrid combination of the previous two. Probably the most representative example here is "chamfer crossing" in a peg-in-hole task: before the proper peg insertion task (as described in the previous Chapter) can be started, the bottom of the peg has first to cross the chamfer of the hole.

Van Brussel & Simons (1979), and Simons & al. (1982) are early and much cited references that deal with, among other things, the chamfer crossing problem. They automatically cluster the force readings in a limited set of classes. From each class an a priori specified action brings the peg closer to the entry chamfer.

Ahmad & Lee (1990) describe a force controlled 2D contour following experiment. The emphasis is on compensation of joint compliance, gear backlash and gear eccentricity, in order to obtain better shape measurements for the contoured object. The authors apply a crude and —even at the time of publication— not state-of-the-art approach to contour

tracking: they move the robot a small distance, measure the force, estimate the tangent, re-orient the motion direction, and start over again.

Hirai & Iwata (1992) apply the theory of Hirai & Asada (1993), i.e., the "Admissible Displacement Set," to a simple experiment: putting a rectangular block on top of a table. The measured forces are used to identify the contact state: which of the four vertices is in contact with the table. However, this information is not used as feedback to the task controller. Indeed, the sequence of controller actions has been determined off-line, and the measurements are only used to detect state transitions.

Bicchi, Salisbury & Brock (1993) concentrate on the *design* of a contact sensor, based on force sensing only. It describes just one contact type, the *soft finger*, and only first order uncertainties (contact point and normal). The main contribution is the definition of the *contact centroid*, i.e., a generalization of the contact point for soft objects. This concept is useful for robotic hands, but much less so for compliant motion applications. No experiments are reported.

Reynaerts & Van Brussel (1993a) describe how a *tactile array* can be used to derive curvature information of an object pressed between two fingers of a dextrous hand.

Lee & Asada (1994) present an interesting active sensing strategy to cope with very high friction during a "peg-in-hole" assembly. Whenever the peg feels a high insertion resistance, the nominal insertion motion is stopped, and a sinusoidal and transversal *dithering motion* is executed. The resulting force readings are correlated with the active sensing motion, which gives an indication of the transversal direction in which to move the peg over a predefined small offset. Thereafter the nominal insertion continues. Geometric uncertainties are not explicitly identified in this approach; however, this would be practically infeasible and unnecessary too.

Mimura & Funahashi (1994) present a theoretical approach to identification of a *completely unknown* surface based on force measurements only. Although the authors realize and understand the necessity for *active sensing* when insufficient identification data is available instantaneously, they rely on very unrealistic assumptions —i.e, the contact situation does not change when arbitrary excitations are applied to it— they use a simple but also simplistic representation for the contact as well as the uncertainties —i.e., inaccurate parameters in the homogeneous transformation between the contact frame and the end effector frame of the robot— and, not surprisingly, they do not describe experimental work.

2.6.2 Motion based identification

Stute & Erne (1979) make their robot move some sort of "comb" with extensible fingers over the *ENV*, and the measured finger extensions give an indication about the surface geometry. However, they don't use this information to adapt the robot's motion.

Asada & Hanafusa (1980), Hanafusa & al. (1981), and Wampler (1984) are some early references that use first order identification and tracking algorithms to adapt the robot's motion specification. These references are very often overlooked. The first two because they are very difficult to read and understand; the last one because of its misleading title.

Nevertheless, they presented viable solutions for contour or surface following that are still applicable today.

The above-mentioned references from the area of manipulator calibration both use motion based identification: the manipulator executes some off-line specified motions, and the differences between these nominally expected motions and the really measured motions serve as inputs to a batch identification procedure. Classically, this kind of papers applies "pseudo-inverse"-like procedures.

2.6.3 Force and motion based identification

Lu, Duffie & Bollinger (1982) apply a force/motion identification strategy to the control of a Coordinate Measurement Machine, in order to generate automatically measurement paths for tracing of unknown workpieces. "Force sensing" is used in on/off mode only, i.e., to determine the point in space where the transition occurs between no contact and contact, or vice versa.

Blauer & Bélanger (1987) describe force and motion based identification, for one single *vertex-surface* contact. The surface is described analytically, with "some unknown parameters (...) Reasonable estimates of the unknown parameters are also available." The authors state the problem of this thesis as the "simultaneous state estimation and parameter identification" of a dynamical system. They apply Extended Kalman Filtering to do this estimation from the noisy process data. They also assume the noise characteristics known. The material of this paper is complementary to the work described in this thesis: Blauer and Bélanger start where this thesis stops, i.e., at the estimation of the uncertainties. However, only simulation results are presented.

De Schutter & Van Brussel (1988a) present identification algorithms for orientation uncertainties. The principles of their force and velocity based identification are shown in FIG. 2.1: the angular uncertainty $\Delta\alpha$ is estimated as the arc tangent of, respectively: 1) the normal component of the measured velocity over the tangential component, or 2) the tangential component of the measured force over the normal component. The measurements are related to the current position of the task frame, indicated by the t subscript in FIG. 2.1; the o subscript indicate the real tangent and normal directions.

De Schutter (1988) presents advanced uncertainty tracking, by means of an observer of the *time behaviour* of the identification, and not only the instantaneous identification data. For the example of opening a door, this results in estimation of the distance to the door hinge. In fact, this is a special case of tracking of a second order uncertainty.

Kazanzides, Bradley & Wolovich (1989) present first order identification and tracking, identical to the earlier paper by De Schutter & Van Brussel (1988a), but without referencing it.

Yoshikawa & Sudou (1993) describe a surface following experiment (restricted to a planar cross-section of the 3D world), with first order identification: the contact "normal" direction (i.e., its projection on the above-mentioned cross-section) is derived from the measured force, from which the component in the motion direction is subtracted. In their

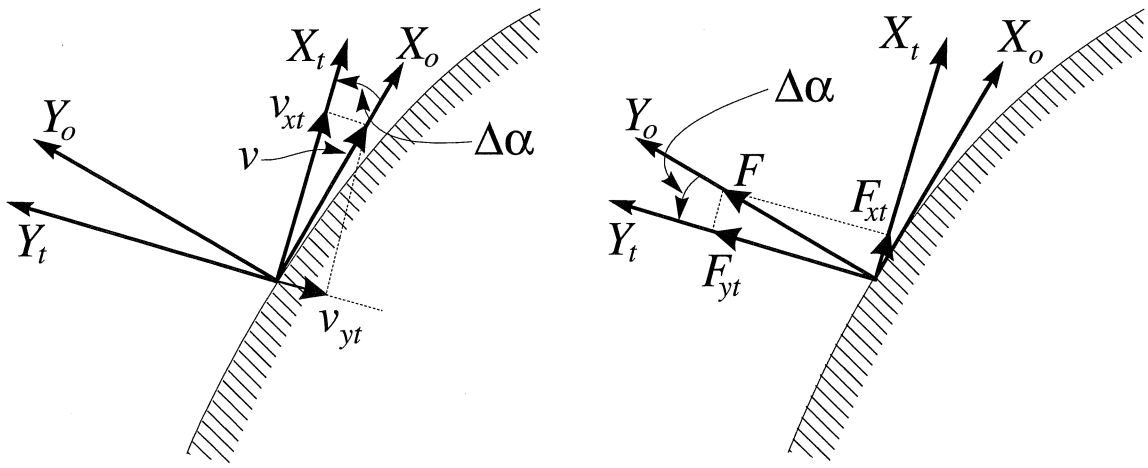


FIGURE 2.1 : *2D estimation of the normal.* The left-hand side shows the principles of identification based on force measurement; the right-hand side is velocity based identification.

opinion, a full dynamic model of the robot system is required to do identification and tracking.

McCarragher & Asada (1993) and McCarragher (1993) concentrate on one particular and rather difficult assembly task —dual peg-in-hole in 2.5D— for which they develop a detailed ad hoc dynamic model and an execution *monitor* based on Petri nets. The task execution goes through a number of different contact states, which are identified by the dynamically interpreted force and motion readings. These have been clustered beforehand in meaningful regions that use the same motion/force specification to continue to the goal state. No general model or identification algorithms are given.

2.7 Conclusions

The literature survey above leads to the following conclusions, which strongly motivate the relevance of this thesis:

- Until now, no *integration* between modelling, identification and sensing has been reported.
- Identification of motion constraints is limited to intuitive *first order strategies*, without general theoretical foundations.
- *Second order motion specifications* exist, but in a form that is not very practical to work with when the uncertainties cannot be sensed directly but have to be observed from motion and force measurements at the robot end effector.
- No experimental work on identification of *composite constraints* is reported.

- No *second order identification* algorithms are reported.
- Invariant descriptions of force controlled compliant motion are still very rare.

Chapter 3

Jacobians and their derivatives

3.1 Introduction

This Chapter introduces the terminology and definitions used in the following Chapters to represent motion of a rigid body, and reaction forces exerted on the body. The instantaneous first order motion of a rigid body is uniquely defined physically. It is called the instantaneous velocity (or *twist* t, Ball (1900)), or, if time is not taken into consideration, the infinitesimal displacement (or *infinitesimal displacement twist* t_Δ). Mathematically, however, the instantaneous motion can be represented in different ways: by choosing different reference points on the rigid body, and by expressing the motion with respect to different "world" frames. The physical and mathematical description for *wrenches* (i.e., forces and torques working on a rigid body) is completely similar to the description for twists.

As already mentioned in the introductory Chapter, one of the main ideas in this thesis is to systematically model a contact between rigid bodies by means of a virtual contact manipulator. From the mathematical point of view, describing the motion characteristics of a virtual contact manipulator is completely similar to the description of a "normal" kinematic chain such as an industrial robot arm. Hence, basic terminology and definitions of robot kinematics can be used. Nevertheless, many kinematic concepts and algorithms needed for the modelling of geometric uncertainties in the motion constraint models, as well as for their on-line identification, have not yet been described in the robotics or kinematics literature. The main "missing links," for which this thesis has developed original contributions, are: 1) the definitions of, and transformations between, the most common —but often only implicitly defined!— mathematical representations of a kinematic chain's *Jacobian matrices*, which describe the mapping from joint motions to Cartesian motions of the chain's end effector; and 2) the calculation of the derivatives of these Jacobians with respect to a change in one of the chain's joint angles. This thesis requires the Jacobian derivatives for two purposes:

1. Motion specification: the velocity t of a kinematic chain's end effector depends linearly on the joint velocities $\dot{\boldsymbol{q}}$ of the chain: $\mathsf{t} = \boldsymbol{J}\dot{\boldsymbol{q}}$. The time derivative of this

equation, i.e., $\dot{\mathbf{t}} = \boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}$, gives the end effector acceleration as a function of the joint velocities and accelerations.

2. Uncertainty modelling: the Jacobian $\boldsymbol{J}$ of a virtual manipulator represents the instantaneous motion freedom of the *MO*. This motion freedom is not exactly known, since some of the geometric parameters of the contact are uncertain. Mathematically, this is represented by the dependence of the Jacobian on a set of uncertainty parameters υ_i: $\boldsymbol{J} = \boldsymbol{J}(\upsilon_i)$. Chapter 5 explains how to model these uncertainty parameters as joints in the virtual manipulators. Hence, the influence of an uncertainty υ_i on the instantaneous motion freedom of the *MO* is given by $\partial \boldsymbol{J}/\partial \upsilon_i$.

A significant part of this Chapter contains tedious but necessary algebraic manipulations. These can be skipped, however, without loosing hold of the main ideas.

Section 3.2 discusses the different representations for twists and wrenches, as well as their mutual transformations. Only a very concise summary is given, with no more than the material needed by the following Sections.

Section 3.3 applies this material to a kinematic chain's Jacobian matrix, and explains how to formally calculate this Jacobian.

Section 3.4 contains the most original material in this Chapter: it defines the derivative of a Jacobian matrix with respect to a change in one of its joint angles, and it also develops symbolic formulas to find this derivative from the original Jacobian itself. The material is developed for a general manipulator with only revolute joints, but with simple modifications it is adapted to manipulators containing one or more prismatic joints.

Section 3.5 illustrates the correctness of the symbolic formulas by the example of a very common kinematic chain for industrial robot arms.

The Appendices A and B apply the main results from this Chapter to the examples of the *surface-surface* and *curve-curve* contacts, and explain how to interpret these results geometrically.

3.2 Twists and wrenches

Consider a rigid body moving with a certain velocity with respect to a reference frame. This physical rigid body motion is called a *twist*. A mathematically unambiguous representation requires the description of: 1) the angular velocity of the body with respect to the reference frame, and 2) the linear velocity of a *reference point* in the rigid body, as seen from the same reference frame, Angeles (1988), Bottema & Roth (1979). Each of these motion components is a three-vector. The combination of the angular velocity and the velocity of a reference point is a six-vector. With a slight abuse of nomenclature, these six numbers are also called the twist of the rigid body. The combination of the angular acceleration and the acceleration of a reference point is another six-vector, called the twist rate, Angeles (1993). A static force on a rigid object is described by a linear force three-vector acting in a point on the rigid body, together with a moment three-vector. The combination of both

is called a *wrench*. Several mathematical representations for twists, wrenches and twist rates exist. The distinctions depend on:

1. The choice of how to combine the two three-vectors into one six-vector, Section 3.2.1.
2. The choice of the "world" (or "base") frame, with respect to which the three-vectors are expressed, Section 3.2.2.
3. The choice of the "velocity reference point," i.e., that point on the rigid body whose velocity is taken as the linear velocity three-vector of the twist, Section 3.2.3.

Section 3.2.4 describes the very common representation in which the reference point instantaneously coincides with the origin of the world frame.

3.2.1 Choice of six-vector

This text consistently uses the following convention for a twist six-vector t and a wrench six-vector w:

$$\mathsf{t} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}, \quad \mathsf{w} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{m} \end{bmatrix}. \tag{3.1}$$

$\boldsymbol{\omega}$ is the angular velocity three-vector of the rigid body; $\boldsymbol{v}$ is the velocity three-vector of the reference point on the rigid body. (The alternative choice, $\mathsf{t} = (\boldsymbol{v}^T \ \boldsymbol{\omega}^T)^T$, is also in widespread use.) $\boldsymbol{f}$ is a linear force three-vector; $\boldsymbol{m}$ is a moment three-vector. If it is necessary to explicitly indicate that the components of t are expressed in a reference frame $\{i\}$ then a leading subscript "i" is added: ${}_i\mathsf{t}$. If it is necessary to explicitly mention the velocity reference point on the rigid body —for example, the origin of the frame $\{j\}$— a trailing superscript "j" is added: t^j. A twist rate six-vector is the time derivative of a twist:

$$\dot{\mathsf{t}} = \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{v}} \end{bmatrix}. \tag{3.2}$$

3.2.2 Choice of reference frame

The left-hand side of FIG. 3.1 shows two reference frames $\{f\}$ and $\{g\}$ fixed to the world, and a reference frame $\{m\}$ fixed to the moving rigid body. The origin of $\{m\}$ serves as the velocity reference point. Let the twist of the rigid body with respect to the frames $\{f\}$ and $\{g\}$ be given by the six-vectors

$${}_f\mathsf{t} = \begin{bmatrix} {}_f\boldsymbol{\omega}^m \\ {}_f\boldsymbol{v}^m \end{bmatrix}, \quad {}_g\mathsf{t} = \begin{bmatrix} {}_g\boldsymbol{\omega}^m \\ {}_g\boldsymbol{v}^m \end{bmatrix}.$$

(The trailing superscript "m" will be omitted for the rest of this paragraph since the velocity reference point remains unchanged.) Physically, the angular velocity vector $\boldsymbol{\omega}$ is the same for both $\{f\}$ and $\{g\}$, as well as the velocity vector $\boldsymbol{v}$. However, their mathematical representations as three numbers, being the projections of the physical vector on the axes

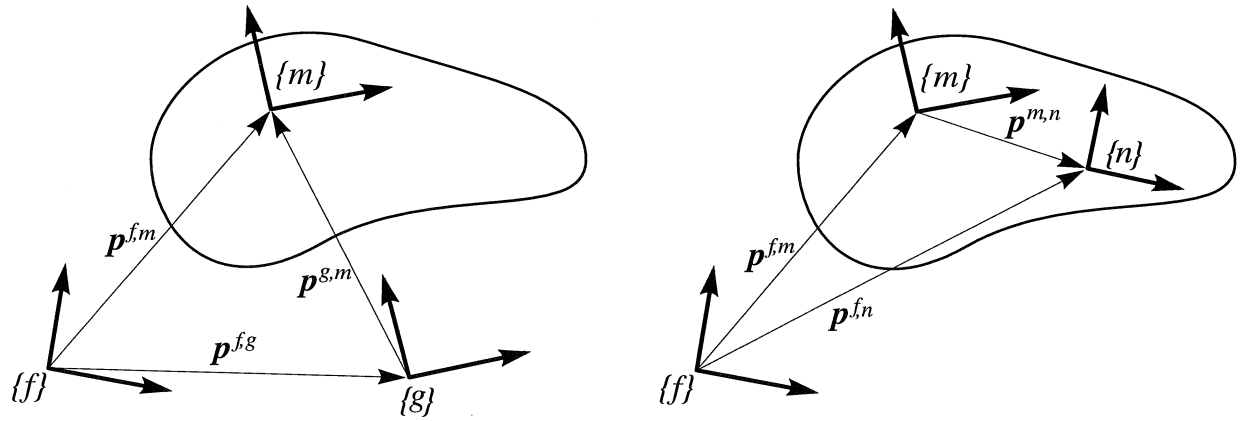


FIGURE 3.1 : *Geometric parameters for transformation of twists.* $\{f\}$ and $\{g\}$ are reference frames fixed to the world. $\{m\}$ and $\{n\}$ are reference frames fixed to the moving rigid body. The vectors $\boldsymbol{p}^{i,j}$ link the origin of frame $\{i\}$ to the origin of frame $\{j\}$.

of both reference frames, are not, because both frames are in general not parallel. These components are linked by the rotation matrix ${}^g_f\boldsymbol{R}$ between both frames:

$$\begin{bmatrix} {}_f\omega_x \\ {}_f\omega_y \\ {}_f\omega_z \end{bmatrix} = {}^g_f\boldsymbol{R} \begin{bmatrix} {}_g\omega_x \\ {}_g\omega_y \\ {}_g\omega_z \end{bmatrix} \qquad \begin{bmatrix} {}_fv_x \\ {}_fv_y \\ {}_fv_z \end{bmatrix} = {}^g_f\boldsymbol{R} \begin{bmatrix} {}_gv_x \\ {}_gv_y \\ {}_gv_z \end{bmatrix}. \tag{3.3}$$

A column in the 3×3 rotation matrix ${}^g_f\boldsymbol{R}$ contains the components of a unit vector along the X, Y and Z axes of frame $\{g\}$, as seen from frame $\{f\}$. The total transformation of the twist, from $\{g\}$ to $\{f\}$, is collected in a 6×6 matrix ${}^g_f\boldsymbol{P}$, called the *screw projection matrix*. (A *screw* is the geometric entity underlying both twists and wrenches. A wrench is a six-vector containing a linear force three-vector and a torque three-vector. See for example Hunt (1990) for an extensive review on screw theory.) ${}^g_f\boldsymbol{P}$ does not change the physical three-vectors of the twist, but just projects them onto another reference frame; this projection changes the numerical values:

$${}_f\mathsf{t} = {}^g_f\boldsymbol{P}\,{}_g\mathsf{t} = \begin{bmatrix} {}^g_f\boldsymbol{R} & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & {}^g_f\boldsymbol{R} \end{bmatrix} {}_g\mathsf{t}. \tag{3.4}$$

The inverse transformation is simply

$$\left({}^g_f\boldsymbol{P}\right)^{-1} = {}^f_g\boldsymbol{P}.$$

Remark that ${}^g_f\boldsymbol{P}$ does not depend on the choice of velocity reference point. Twist rates and wrenches transform in the same way as twists.

3.2.3 Choice of velocity reference point

The right-hand side of FIG. 3.1 shows two reference points, the origins of the frames $\{m\}$ and $\{n\}$, fixed to the rigid body, and a reference frame $\{f\}$ fixed to the world. Let the twist of the rigid body with the origin of $\{n\}$ as velocity reference point be given by the six-vector

$$ {}_f\mathsf{t}^n = \begin{bmatrix} {}_f\boldsymbol{\omega}^n \\ {}_f\boldsymbol{v}^n \end{bmatrix}. $$

(The leading subscript "f" will be omitted for the rest of this paragraph since all motion parameters are expressed with respect to the same frame $\{f\}$.) The physical angular velocity vector $\boldsymbol{\omega}$ does not change under the transformation of the velocity reference point from the origin of $\{n\}$ to the origin of $\{m\}$, but the physical translational velocity vector $\boldsymbol{v}$ does:

$$ \begin{aligned} \boldsymbol{\omega}^m &= \boldsymbol{\omega}^n \quad (\triangleq \boldsymbol{\omega}), \\ \boldsymbol{v}^m &= \boldsymbol{v}^n + \boldsymbol{p}^{m,n} \times \boldsymbol{\omega}, \end{aligned} \tag{3.5} $$

with $\boldsymbol{p}^{m,n}$ the vector from the origin of frame $\{m\}$ to the origin of the frame $\{n\}$. Hence, the velocity components transform as

$$ \begin{bmatrix} v_x^m \\ v_y^m \\ v_z^m \end{bmatrix} = \begin{bmatrix} v_x^n \\ v_y^n \\ v_z^n \end{bmatrix} + \begin{bmatrix} 0 & -p_z^{m,n} & p_y^{m,n} \\ p_z^{m,n} & 0 & -p_x^{m,n} \\ -p_y^{m,n} & p_x^{m,n} & 0 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \tag{3.6} $$

And the twist t transforms with the following so-called *reference point transformation matrix* $\boldsymbol{M}_n^m$:

$$ \mathsf{t}^m = \boldsymbol{M}_n^m \mathsf{t}^n = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{O}_{3\times3} \\ [\boldsymbol{p}^{m,n}\times] & \boldsymbol{I}_{3\times3} \end{bmatrix} \mathsf{t}^n. \tag{3.7} $$

$[\boldsymbol{p}^{m,n}\times]$ is the 3×3 skew-symmetric matrix representing the cross product with $\boldsymbol{p}^{m,n}$, see Eq. (3.6). $\boldsymbol{M}_n^m$ changes the twist's physical linear velocity three-vector (and hence also its components), but leaves the physical angular velocity three-vector unchanged, as well as its components. Again, the inverse transformation is simply

$$ \left(\boldsymbol{M}_n^m\right)^{-1} = \boldsymbol{M}_m^n. $$

If the world frame has to be indicated explicitly, a leading subscript is added: ${}_f\boldsymbol{M}_n^m$. Wrenches transform in the same way as twists.

Let the twist rate six-vector of the rigid body with respect to $\{f\}$, and with $\{n\}$ as reference on the body, be given by:

$$ \dot{\mathsf{t}}^n = \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{v}}^n \end{bmatrix}. $$

The time derivative of Eq. 3.5 yields:

$$\begin{aligned}\dot{\boldsymbol{v}}^m &= \frac{d}{dt}\left(\boldsymbol{v}^n + \boldsymbol{p}^{m,n} \times \boldsymbol{\omega}\right) \\ &= \dot{\boldsymbol{v}}^n + \dot{\boldsymbol{p}}^{m,n} \times \boldsymbol{\omega} + \boldsymbol{p}^{m,n} \times \dot{\boldsymbol{\omega}}.\end{aligned}$$

Since both frames $\{m\}$ and $\{n\}$ move with respect to $\{f\}$, $\dot{\boldsymbol{p}}^{m,n}$ is not zero as seen from $\{f\}$:

$$\begin{aligned}\dot{\boldsymbol{p}}^{m,n} &= \dot{\boldsymbol{p}}^{f,n} - \dot{\boldsymbol{p}}^{f,m} \\ &= \boldsymbol{v}^n - \boldsymbol{v}^m \\ &= \boldsymbol{v}^n - \left(\boldsymbol{v}^n + \boldsymbol{p}^{m,n} \times \boldsymbol{\omega}\right) \\ &= -\boldsymbol{p}^{m,n} \times \boldsymbol{\omega}.\end{aligned}$$

This yields a *non-linear* transformation from $\dot{\mathsf{t}}^n$ to $\dot{\mathsf{t}}^m$:

$$\dot{\boldsymbol{v}}^m = \dot{\boldsymbol{v}}^n - \left(\boldsymbol{p}^{m,n} \times \boldsymbol{\omega}\right) \times \boldsymbol{\omega} + \boldsymbol{p}^{m,n} \times \dot{\boldsymbol{\omega}}. \tag{3.8}$$

Notational remark: the screw projection matrix ${}^g_f\boldsymbol{P}$ has leading indices, while the reference point transformation matrix $\boldsymbol{M}^m_n$ has trailing indices. The reason for this convention is clear from Eqs. (3.4) and (3.7): ${}^g_f\boldsymbol{P}$ "acts" on the leading subscript of a twist, and $\boldsymbol{M}^m_n$ on the trailing superscript, according to a simple index replacement scheme similar to Einstein's summation convention.

3.2.4 Velocity reference point coincides with world frame origin

In order to reduce the bookkeeping of reference points and frames, twists are very often defined as follows: the velocity reference point on the rigid body coincides with the origin of the world frame. (This coincidence is only *instantaneous*, i.e., the velocity reference point moves with respect to the moving rigid body.) Notationally, this means that the leading subscript equals the trailing superscript. The transformation of a twist t from a frame $\{g\}$ to a frame $\{f\}$ is easily derived from the previous paragraphs, Eqs. (3.4) and (3.7). The corresponding transformation matrix is called the *screw transformation* matrix, ${}^g_f\boldsymbol{S}$. (Angeles (1994) calls it the *twist-transfer* matrix. In the kinematics and dynamics literature inspired by differential geometry, Lie groups and Lie algebras, this transformation is the *adjoint operator*, Karger & Novak (1985), Park & Bobrow (1994).) It works as

$${}_f\mathsf{t}^f = {}^g_f\boldsymbol{S}\,{}_g\mathsf{t}^g, \tag{3.9}$$

$$\begin{aligned}\text{with}\quad {}^g_f\boldsymbol{S} &= \begin{bmatrix} {}^g_f\boldsymbol{R} & \boldsymbol{O}_{3\times3} \\ \left[{}_f\boldsymbol{p}^{f,g}\times\right] {}^g_f\boldsymbol{R} & {}^g_f\boldsymbol{R} \end{bmatrix} = {}_f\boldsymbol{M}^f_g\ {}^g_f\boldsymbol{P} \\ &= \begin{bmatrix} {}^g_f\boldsymbol{R} & \boldsymbol{O}_{3\times3} \\ {}^g_f\boldsymbol{R}\left[{}_g\boldsymbol{p}^{f,g}\times\right] & {}^g_f\boldsymbol{R} \end{bmatrix} = {}^g_f\boldsymbol{P}\ {}_g\boldsymbol{M}^f_g.\end{aligned}$$

It changes the physical three-vectors of the twist, as well as the reference frame in which their components are expressed. The inverse transformation is simply

$$\left({}^{g}_{f}\boldsymbol{S}\right)^{-1} = {}^{f}_{g}\boldsymbol{S}.$$

Since the screw transformation ${}^{g}_{f}\boldsymbol{S}$ works on the reference frame as well as on the velocity reference point, it should have leading *and* trailing indices. However, these indices would, by definition, always be identical, so the trailing ones are omitted.

3.2.5 Remark

The robotics literature often defines another twist six-vector convention, i.e., with the positions of $\boldsymbol{\omega}$ and $\boldsymbol{v}$ interchanged:

$$\tilde{\mathsf{t}} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = \tilde{\boldsymbol{\Delta}}\,\mathsf{t}, \text{ with } \tilde{\boldsymbol{\Delta}} \triangleq \begin{bmatrix} \boldsymbol{O}_{3\times3} & \boldsymbol{I}_{3\times3} \\ \boldsymbol{I}_{3\times3} & \boldsymbol{O}_{3\times3} \end{bmatrix}. \tag{3.10}$$

The difference may seem meaningless at first sight, but it certainly is not: the stiffness matrix $\boldsymbol{K}$, the damping matrix $\boldsymbol{C}$ and the generalized inertia matrix $\boldsymbol{I}$, which all map motion-type of six-vectors (displacement, velocity or acceleration) into a wrench, have more physical meaning in the "t" representation than in the "$\tilde{\mathsf{t}}$" representation. The reason is that the linear algebraic concepts of eigenvalues and eigenvectors have physical meaning for $\boldsymbol{K}$, $\boldsymbol{C}$ and $\boldsymbol{I}$ but not for $\tilde{\boldsymbol{K}}$, $\tilde{\boldsymbol{C}}$ and $\tilde{\boldsymbol{I}}$. This can be seen from their transformation formulae, deduced from Eq. (3.9). Let ${}_{g}\boldsymbol{K}$ be a stiffness matrix, expressed with respect to the reference frame $\{g\}$, and ${}_{g}\mathsf{t}_{\Delta}$ an infinitesimal twist representing the deformation of the stiffness. Hence, the resulting wrench is calculated as:

$${}_{g}\mathsf{w} = {}_{g}\boldsymbol{K}\,{}_{g}\mathsf{t}_{\Delta}.$$

Transformation to the reference frame $\{f\}$ yields:

$${}_{f}\mathsf{w} = {}^{g}_{f}\boldsymbol{S}\,{}_{g}\mathsf{w} = {}^{g}_{f}\boldsymbol{S}\,{}_{g}\boldsymbol{K}\left({}^{g}_{f}\boldsymbol{S}\right)^{-1}{}_{f}\mathsf{t}_{\Delta},$$

or:

$${}_{f}\boldsymbol{K} = {}^{g}_{f}\boldsymbol{S}\,{}_{g}\boldsymbol{K}\left({}^{g}_{f}\boldsymbol{S}\right)^{-1}.$$

Hence, the stiffness matrix $\boldsymbol{K}$ transforms with a *similarity* transformation, which leaves the eigenvectors and eigenvalues unchanged, Strang (1988). Doing the same calculations for $\tilde{\boldsymbol{K}}$ gives:

$$\begin{aligned}
{}_{f}\mathsf{w} &= {}^{g}_{f}\boldsymbol{S}\,{}_{g}\mathsf{w} \\
&= {}^{g}_{f}\boldsymbol{S}\,{}_{g}\tilde{\boldsymbol{K}}\,{}_{g}\tilde{\mathsf{t}}_{\Delta} \\
&= {}^{g}_{f}\boldsymbol{S}\,{}_{g}\tilde{\boldsymbol{K}}\,\tilde{\boldsymbol{\Delta}}\,{}_{g}\mathsf{t}_{\Delta} \\
&= {}^{g}_{f}\boldsymbol{S}\,{}_{g}\tilde{\boldsymbol{K}}\,\tilde{\boldsymbol{\Delta}}\left({}^{g}_{f}\boldsymbol{S}\right)^{-1}{}_{f}\mathsf{t}_{\Delta} \\
&= {}^{g}_{f}\boldsymbol{S}\,{}_{g}\tilde{\boldsymbol{K}}\,\tilde{\boldsymbol{\Delta}}\left({}^{g}_{f}\boldsymbol{S}\right)^{-1}\tilde{\boldsymbol{\Delta}}\,{}_{f}\tilde{\mathsf{t}}_{\Delta} \\
&= {}^{g}_{f}\boldsymbol{S}\,{}_{g}\tilde{\boldsymbol{K}}\left({}^{g}_{f}\boldsymbol{S}\right)^{T}{}_{f}\tilde{\mathsf{t}}_{\Delta},
\end{aligned}$$

because

$$ {}^{g}_{f}\boldsymbol{S} = \begin{bmatrix} {}^{g}_{f}\boldsymbol{R} & \boldsymbol{O} \\ [\boldsymbol{p}\times]\,{}^{g}_{f}\boldsymbol{R} & {}^{g}_{f}\boldsymbol{R} \end{bmatrix}, \quad \left({}^{g}_{f}\boldsymbol{S}\right)^{-1} = \begin{bmatrix} \left({}^{g}_{f}\boldsymbol{R}\right)^{T} & \boldsymbol{O} \\ -\left({}^{g}_{f}\boldsymbol{R}\right)^{T}[\boldsymbol{p}\times] & \left({}^{g}_{f}\boldsymbol{R}\right)^{T} \end{bmatrix} = \widetilde{\boldsymbol{\Delta}}\,\left({}^{g}_{f}\boldsymbol{S}\right)^{T}\,\widetilde{\boldsymbol{\Delta}}. $$

Since $\left(\widetilde{\boldsymbol{\Delta}}\right)^{-1} = \widetilde{\boldsymbol{\Delta}}$:

$$ \widetilde{\boldsymbol{\Delta}}\,\left({}^{g}_{f}\boldsymbol{S}\right)^{-1}\widetilde{\boldsymbol{\Delta}} = \left({}^{g}_{f}\boldsymbol{S}\right)^{T}. $$

And:

$$ {}_{f}\widetilde{\boldsymbol{K}} = {}^{g}_{f}\boldsymbol{S}\;{}_{g}\widetilde{\boldsymbol{K}}\left({}^{g}_{f}\boldsymbol{S}\right)^{T}. $$

This means that $\widetilde{\boldsymbol{K}}$ transforms with a *congruence* transformation, Strang (1988), Griffis & Duffy (1991), which does not preserve eigenvectors and eigenvalues, but only the *signature* of the matrix, i.e., the number of positive, zero and negative eigenvalues! This fact is another important source of non-invariance in current robotics research.

3.3 Jacobian matrix for a serial chain

Similar to the definition of the *Jacobian matrix* for a mathematical function of several variables, the Jacobian matrix of a serial kinematic chain is defined as the "matrix of partial derivatives" of the chain's end effector Cartesian position with respect to its joint angles, Whitney (1969). However, the orientation part of the end effector position does not have a well-defined time derivative, Haug (1989), regardless of the mathematical representation! Hence, the kinematic Jacobian matrix cannot be calculated by straightforward derivation of the chain's forward position kinematics, and thus it is *not* a mathematical Jacobian. Nevertheless, the kinematic Jacobian is still a linear mapping: for a manipulator with n joints, this mapping is given by a $6 \times n$ matrix $\boldsymbol{J}$. Note however that this mapping depends —non-linearly!— on the joint positions: $\boldsymbol{J} = \boldsymbol{J}(\boldsymbol{q})$. $\boldsymbol{J}^{i}$, i.e., column i of this Jacobian $\boldsymbol{J}$, represents the Cartesian twist of the end effector due to joint velocity $\dot{q}^{i}$ of joint i. Linearity means that applying a velocity to all joints results in a twist t that is simply the sum of the contributions from all joints:

$$ \mathsf{t} = \begin{bmatrix} \boldsymbol{J}^{1} & \dots & \boldsymbol{J}^{n} \end{bmatrix} \begin{bmatrix} \dot{q}^{1} \\ \vdots \\ \dot{q}^{n} \end{bmatrix} = \boldsymbol{J}\dot{\boldsymbol{q}}, \tag{3.11} $$

with $\dot{\boldsymbol{q}} = (\dot{q}^{1} \dots \dot{q}^{n})^{T}$ the vector of joint velocities. Several algorithms have been presented to calculate the Jacobian, e.g., Craig (1986), Driels (1986), Orin & Schrader (1984), Paul (1986), Whitney (1972). These algorithms require only the knowledge of the current geometrical parameters, such as, for example, contained in the Denavit-Hartenberg parameters, Denavit & Hartenberg (1955).

This Section introduces the "body-fixed," "inertial" and "hybrid" representations for Jacobian matrices. It also repeats the well-known formal construction of the Jacobian

	inertial	body-fixed	hybrid
inertial		${}_{ee}\mathsf{t}^{ee} = {}^{bs}_{ee}\boldsymbol{S}\ {}_{bs}\mathsf{t}^{bs}$	${}_{bs}\mathsf{t}^{ee} = \boldsymbol{M}^{ee}_{bs}\ {}_{bs}\mathsf{t}^{bs}$
body-fixed	${}_{bs}\mathsf{t}^{bs} = {}^{ee}_{bs}\boldsymbol{S}\ {}_{ee}\mathsf{t}^{ee}$		${}_{bs}\mathsf{t}^{ee} = {}^{ee}_{bs}\boldsymbol{P}\ {}_{ee}\mathsf{t}^{ee}$
hybrid	${}_{bs}\mathsf{t}^{bs} = \boldsymbol{M}^{bs}_{ee}\ {}_{bs}\mathsf{t}^{ee}$	${}_{ee}\mathsf{t}^{ee} = {}^{bs}_{ee}\boldsymbol{P}\ {}_{bs}\mathsf{t}^{ee}$	

TABLE 3.1 : *Transformation between representations.* Each entry in this table indicates which transformation matrix to use in order to transform a twist in the representation of the left-hand side into the representation on the top of the table.

and its time derivative —in each of the representations— starting from the geometric parameters of the manipulator in a given position.

3.3.1 Representations

In general, a serial kinematic chain is given (at least) two reference frames, FIG. 3.2: 1) a "base" frame, $\{bs\}$, rigidly fixed to the world, i.e., the immobile base of the chain, and 2) the "end effector" frame, $\{ee\}$, rigidly fixed to the last link of the chain. These two frames give rise to the following "natural" representations of the end effector twist:

Body-fixed. In the body-fixed representation, the end effector frame $\{ee\}$ serves also as reference frame and its origin as the velocity reference point. Hence, the full notation of a twist in this representation is ${}_{ee}\mathsf{t}^{ee}$. (Remark that in this representation the reference frame is *not* an absolute inertial frame!)

Inertial. The frame $\{bs\}$ is chosen as the reference frame and its origin as the (instantaneous) velocity reference point. Hence, a twist in this representation is denoted ${}_{bs}\mathsf{t}^{bs}$.

Hybrid. $\{bs\}$ is the reference frame and the origin of $\{ee\}$ is the velocity reference point. Hence, a twist in this representation is denoted ${}_{bs}\mathsf{t}^{ee}$.

Equations (3.4), (3.7) and (3.9) perform the transformations between the different representations, Table 3.1. Since each column of a Jacobian matrix $\boldsymbol{J}$ is a twist, the same set of representations and representation transformations exist for Jacobians. So, the symbol ${}_{bs}\boldsymbol{J}^{j,bs}$ denotes the jth column of $\boldsymbol{J}$ in the inertial representation. Similarly, ${}_{ee}\boldsymbol{J}^{j,ee}$ is the jth column in body-fixed representation, and ${}_{bs}\boldsymbol{J}^{j,ee}$ is the jth column in hybrid representation. If an index is lacking, the corresponding representation convention is assumed to be irrelevant.

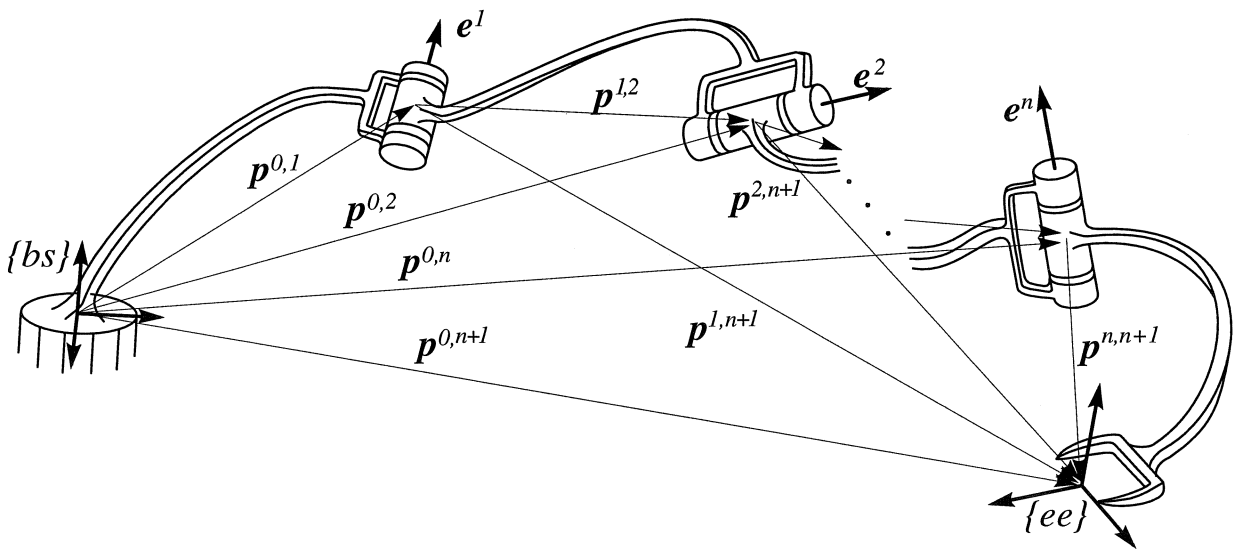


FIGURE 3.2 : *Geometric parameters of a serial kinematic chain.* The calculation of the Jacobian needs the direction vectors $\boldsymbol{e}^i$ of the joint axes, and the position vectors $\boldsymbol{p}^{i,j}$ of the joint axes with respect to the chosen reference frames and with respect to each other.

3.3.2 Construction

Consider a serial kinematic chain, with n revolute joints between base and end effector, FIG. 3.2. The base frame $\{bs\}$ is "joint" 0, and the end effector frame $\{ee\}$ is "joint" $n+1$. Each column of the Jacobian matrix is constructed based on the geometry of the manipulator. Let $\boldsymbol{e}^i$ be the unit vector along the rotation axis of joint i, as well as one of the three axes of the orthogonal reference frame $\{i\}$ fixed to joint i. Let $\boldsymbol{p}^{i,j}$ the vector from the origin of $\{i\}$ to the origin of $\{j\}$. Then, the columns $\boldsymbol{J}_i^{i,bs}$ and $\boldsymbol{J}_i^{i,ee}$ are given by, Angeles (1993):

$$\boldsymbol{J}_i^{i,bs} = \begin{bmatrix} \boldsymbol{e}^i \\ \boldsymbol{p}^{bs,i} \times \boldsymbol{e}^i \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{e}^i \\ \boldsymbol{v}^{i,bs} \end{bmatrix}, \tag{3.12}$$

$$\boldsymbol{J}_i^{i,ee} = \begin{bmatrix} \boldsymbol{e}^i \\ -\boldsymbol{p}^{i,ee} \times \boldsymbol{e}^i \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{e}^i \\ \boldsymbol{v}^{i,ee} \end{bmatrix}. \tag{3.13}$$

The topmost three-vector in these equations represents a unit angular velocity about the axis of joint i. The lower-most three-vector is the translational velocity of the velocity reference point, due to this unit angular velocity. The position vectors $\boldsymbol{p}^{bs,i}$ and $-\boldsymbol{p}^{i,ee} = \boldsymbol{p}^{ee,i}$ are the lever arms from the velocity reference points to the rotating joint. Remark that these equations involve only the physical three-vectors. The choice of reference frame in which their components are expressed does not influence the validity of the equations. Of course, this choice does change the numerical values of the vector components.

The expressions for a column in the twist Jacobian are not limited to *serially connected* chains: only the *instantaneous* and *relative location* of the joints in the chain is important

—regardless of how they are interconnected— together with the fact that the contributions from all joints are *superimposed.*

3.3.3 Time derivative

The time derivative of a Jacobian, denoted $\dot{\boldsymbol{J}}$ or $\partial \boldsymbol{J}/\partial t$, follows formally from the derivative of Eqs. (3.12) and (3.13), Angeles (1993), Patarinski & Botev (1993). For column i this gives

$$\dot{\boldsymbol{J}}^{i,bs} = \begin{bmatrix} \dot{\boldsymbol{e}}^i \\ \frac{\partial}{\partial t}\left(\boldsymbol{p}^{bs,i} \times \boldsymbol{e}^i\right) \end{bmatrix} \qquad \dot{\boldsymbol{J}}^{i,ee} = \begin{bmatrix} \dot{\boldsymbol{e}}^i \\ \frac{\partial}{\partial t}\left(-\boldsymbol{p}^{i,ee} \times \boldsymbol{e}^i\right) \end{bmatrix}. \tag{3.14}$$

The topmost three-vector in these equations represents the change in orientation of joint axis i, due to the joint angle velocities $\dot{q}^j$ of the previous joints in the serial chain. This means that

$$\dot{\boldsymbol{e}}^i = \boldsymbol{\omega}^i \times \boldsymbol{e}^i \quad \text{with} \quad \boldsymbol{\omega}^i = \sum_{j=1}^{i-1} \dot{q}^j \boldsymbol{e}^j. \tag{3.15}$$

The lower-most three-vector in Eq. (3.14) is the acceleration of the velocity reference point as seen from the reference frame. (In the body-fixed representation, this acceleration is *not* the absolute acceleration of the velocity reference point with respect to an inertial frame!) The time derivative of the vector $\boldsymbol{p}^{i,j}$ is

$$\dot{\boldsymbol{p}}^{i,j} = \sum_{k=i}^{j-1} \dot{\boldsymbol{p}}^{k,k+1} \quad \text{with} \quad \dot{\boldsymbol{p}}^{k,k+1} = \boldsymbol{\omega}^k \times \boldsymbol{p}^{k,k+1}. \tag{3.16}$$

The transformations in Table 3.1 allow to calculate the time derivative in one of the three representations from any of the other representations. However, the time derivatives of the transformation matrices ${}^a_b\boldsymbol{S}$, $\boldsymbol{M}^b_a$ and ${}^a_b\boldsymbol{P}$ are needed. These in turn, use ${}^a_b\dot{\boldsymbol{R}}$ and $\dot{\boldsymbol{p}}^{ba}$. The latter one is derived from Eq. (3.16); the former is well-known in the robotics literature, e.g., Angeles (1988), Bottema & Roth (1979):

$${}^a_b\dot{\boldsymbol{R}} = [{}_b\boldsymbol{\omega}\times]\ {}^a_b\boldsymbol{R}, \tag{3.17}$$

with ${}_b\boldsymbol{\omega}$ the angular velocity of frame $\{a\}$ with respect to frame $\{b\}$, and expressed in frame $\{b\}$.

3.4 Derivative with respect to a joint angle

This Section explains how to obtain symbolic expressions for the derivative of a Jacobian matrix with respect to a change in joint angles. This joint angle derivative leads straightforwardly to the symbolic derivative with respect to time. Firstly, however, the derivatives of the transformation matrices ${}^a_b\boldsymbol{S}$, $\boldsymbol{M}^b_a$ and ${}^a_b\boldsymbol{P}$ are determined.

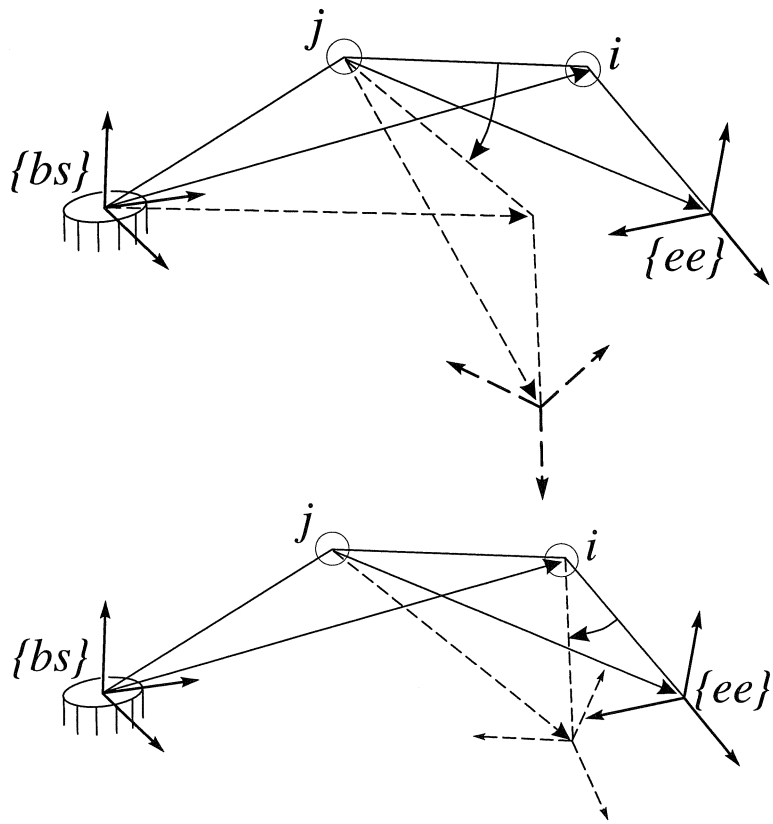


FIGURE 3.3 : *Change of the lever arm between a joint and the base frame* $\{bs\}$ *or the end effector frame* $\{ee\}$. Rotation of a joint j closer to the base than joint i changes the lever arm between joint i and the base. On the other hand, rotating i does not change the lever arm between joint j and the base. Rotation of a joint j closer to the base than joint i does not change the lever arm between joint i and the end effector frame $\{ee\}$, as seen from $\{ee\}$. On the other hand, rotating i does change the lever arm between joint j and $\{ee\}$.

3.4.1 Formal derivation

The derivative of the Jacobian with respect to a joint angle q^j is closely related to its time derivative:

$$\frac{\partial \boldsymbol{J}}{\partial q^j} = \frac{\dot{\boldsymbol{J}}\left(\dot{q}^j \neq 0, \dot{q}^{i\neq j} = 0\right)}{\dot{q}^j} = \dot{\boldsymbol{J}}\left(\dot{q}^j = 1, \dot{q}^{i\neq j} = 0\right). \tag{3.18}$$

The derivatives with respect to time or with respect to a joint angle j make use of the lever arm between joint j and the velocity reference point on the end effector. (Remember that, in the inertial representation, this velocity reference point coincides with the origin of the *base* frame!) Rotating a joint j that is "farther" away from the base of

	inertial	body-fixed	hybrid
	$\frac{\partial\ {}_{bs}\boldsymbol{J}^{i,bs}}{\partial q^j}$	$\frac{\partial\ {}_{ee}\boldsymbol{J}^{i,ee}}{\partial q^j}$	$\frac{\partial\ {}_{bs}\boldsymbol{J}^{i,ee}}{\partial q^j}$
$j = i$	$\equiv 0$	$\equiv 0$	$\neq 0$
$j > i$	$\equiv 0$	$\neq 0$	$\neq 0$
$j < i$	$\neq 0$	$\equiv 0$	$\neq 0$
j n.c. i	$\equiv 0$	$\equiv 0$	$\equiv 0$

TABLE 3.2 : *Derivatives of a column in the Jacobian.* If the derivative of the ith column of $\boldsymbol{J}$ is not zero in the inertial representation, it is identically zero in the body-fixed representation, and vice versa. In both representations, the derivative of a column with respect to its own joint angle is always identically zero. The "hybrid" representation of the third column never gives, in general, identically zero derivatives. "n.c." in the last row means that joints i and j are not connected.

the kinematic chain than joint i does not change the lever arm between base and joint i, neither in magnitude nor in direction, FIG. 3.3. Hence, by the definitions (3.12) and (3.13), the derivative $\partial \boldsymbol{J}^i / \partial q^j$ vanishes in the inertial representation, but not in the body-fixed or hybrid representations. A similar reasoning applies to a joint that is "closer" to the base, FIG. 3.3: body-fixed derivatives vanish, inertial derivatives don't. In the "hybrid" representation, i.e., ${}_{bs}\boldsymbol{J}^{ee}$, the lever arm between base and end effector changes, in general, for all combinations of i and j. Table 3.2 summarizes these properties. Remark also that the derivatives vanish, by definition, if joints i and j are not connected!

3.4.2 Partial derivatives for R

Equations (3.17) and (3.18) applied to the rotation matrix between $\{ee\}$ and $\{bs\}$ gives

$$\frac{\partial {}^{ee}_{bs}\boldsymbol{R}}{\partial q^j} = \left[{}_{bs}\boldsymbol{e}^j \times\right] {}^{ee}_{bs}\boldsymbol{R}, \tag{3.19}$$

since only motion of joint j is to be considered. The identity ${}^{ee}_{bs}\boldsymbol{R}\ {}^{bs}_{ee}\boldsymbol{R} = \boldsymbol{I}$ leads to

$$\frac{\partial {}^{ee}_{bs}\boldsymbol{R}}{\partial q^j} {}^{bs}_{ee}\boldsymbol{R} + {}^{ee}_{bs}\boldsymbol{R} \frac{\partial {}^{bs}_{ee}\boldsymbol{R}}{\partial q^j} = 0,$$

or

$$\frac{\partial {}^{bs}_{ee}\boldsymbol{R}}{\partial q^j} = -\,{}^{bs}_{ee}\boldsymbol{R} \left[{}_{bs}\boldsymbol{e}^j \times\right] = -\left[{}_{ee}\boldsymbol{e}^j \times\right] {}^{bs}_{ee}\boldsymbol{R}. \tag{3.20}$$

3.4.3 Partial derivatives for P

Equations (3.4) and (3.19) yield

$$\frac{\partial {}^{ee}_{bs}\boldsymbol{P}}{\partial q^j} = \begin{bmatrix} [{}_{bs}\boldsymbol{e}^j\times]\,{}^{ee}_{bs}\boldsymbol{R} & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & [{}_{bs}\boldsymbol{e}^j\times]\,{}^{ee}_{bs}\boldsymbol{R} \end{bmatrix} \triangleq \boldsymbol{P}_\Delta\left({}_{bs}\boldsymbol{J}^j\right)\,{}^{ee}_{bs}\boldsymbol{P}, \tag{3.21}$$

with

$$\boldsymbol{P}_\Delta\left({}_{bs}\boldsymbol{J}^j\right) \overset{(3.12)}{=} \boldsymbol{P}_\Delta\left(\begin{bmatrix} {}_{bs}\boldsymbol{e}^j \\ {}_{bs}\boldsymbol{v}^j \end{bmatrix}\right) \triangleq \begin{bmatrix} [{}_{bs}\boldsymbol{e}^j\times] & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & [{}_{bs}\boldsymbol{e}^j\times] \end{bmatrix}. \tag{3.22}$$

The identity ${}^{ee}_{bs}\boldsymbol{P}\;{}^{bs}_{ee}\boldsymbol{P} = \boldsymbol{I}$ leads to

$$\frac{\partial {}^{bs}_{ee}\boldsymbol{P}}{\partial q^j} = -{}^{bs}_{ee}\boldsymbol{P}\;\boldsymbol{P}_\Delta\left({}_{bs}\boldsymbol{J}^j\right). \tag{3.23}$$

$\boldsymbol{P}_\Delta$ depends only on the reference frame, not on the velocity reference point on the end effector. Hence, the trailing superscript is omitted from the $\boldsymbol{J}$'s in Eqs. (3.21)-(3.23).

3.4.4 Partial derivatives for M

By definition of the Jacobian matrix, $\boldsymbol{p}^{bs,ee}$ changes under the influence of a joint rotation Δq^j as, (3.13),

$$\frac{\partial \boldsymbol{p}^{bs,ee}}{\partial q^j} = \boldsymbol{e}^j \times \boldsymbol{p}^{j,ee}. \tag{3.24}$$

Hence

$$\begin{aligned} \frac{\partial \boldsymbol{M}^{bs}_{ee}}{\partial q^j} &\overset{(3.7)}{=} \begin{bmatrix} \boldsymbol{O}_{3\times3} & \boldsymbol{O}_{3\times3} \\ \frac{\partial}{\partial q^j}\left[\boldsymbol{p}^{bs,ee}\times\right] & \boldsymbol{O}_{3\times3} \end{bmatrix} \\ &\overset{(3.13),(3.24)}{=} \begin{bmatrix} \boldsymbol{O}_{3\times3} & \boldsymbol{O}_{3\times3} \\ [\boldsymbol{v}^{j,ee}\times] & \boldsymbol{O}_{3\times3} \end{bmatrix} \\ &\triangleq \boldsymbol{M}_\Delta\left(\boldsymbol{J}^{j,ee}\right). \end{aligned} \tag{3.25}$$

Equations (3.7) and (3.25) imply that

$$\boldsymbol{M}_\Delta\boldsymbol{M} = \boldsymbol{M}_\Delta = \boldsymbol{M}\,\boldsymbol{M}_\Delta. \tag{3.26}$$

This equation and the identity $\boldsymbol{M}^{bs}_{ee}\;\boldsymbol{M}^{ee}_{bs} = \boldsymbol{I}$ lead to

$$\frac{\partial \boldsymbol{M}^{bs}_{ee}}{\partial q^j} = -\frac{\partial \boldsymbol{M}^{ee}_{bs}}{\partial q^j}. \tag{3.27}$$

3.4.5 Partial derivatives for S

A similar reasoning works for ${}^{ee}_{bs}\boldsymbol{S}$:

$$\frac{\partial {}^{ee}_{bs}\boldsymbol{S}}{\partial q^j} = \begin{bmatrix} \frac{\partial {}^{ee}_{bs}\boldsymbol{R}}{\partial q^j} & \boldsymbol{O}_{3\times 3} \\ \frac{\partial [\boldsymbol{p}^{bs,ee}\times]}{\partial q^j} {}^{ee}_{bs}\boldsymbol{R} + [\boldsymbol{p}^{bs,ee}\times] \frac{\partial {}^{ee}_{bs}\boldsymbol{R}}{\partial q^j} & \frac{\partial {}^{ee}_{bs}\boldsymbol{R}}{\partial q^j} \end{bmatrix} \tag{3.28}$$

$$\overset{(3.19),(3.24)}{=} \begin{bmatrix} [\boldsymbol{e}^j\times] {}^{ee}_{bs}\boldsymbol{R} & \boldsymbol{O}_{3\times 3} \\ \left[\left(\boldsymbol{e}^j \times \boldsymbol{p}^{j,bs}\right)\times\right] {}^{ee}_{bs}\boldsymbol{R} + [\boldsymbol{e}^j\times]\left[\boldsymbol{p}^{bs,ee}\times\right] {}^{ee}_{bs}\boldsymbol{R} & [\boldsymbol{e}^j\times] {}^{ee}_{bs}\boldsymbol{R} \end{bmatrix} \tag{3.29}$$

$$\overset{(3.13)}{=} \boldsymbol{S}_\Delta\left({}_{bs}\boldsymbol{J}^{j,bs}\right) {}^{ee}_{bs}\boldsymbol{S}, \tag{3.30}$$

$$\boldsymbol{S}_\Delta\left({}_{bs}\boldsymbol{J}^{j,bs}\right) \triangleq \begin{bmatrix} [{}_{bs}\boldsymbol{e}^j\times] & \boldsymbol{O}_{3\times 3} \\ [{}_{bs}\boldsymbol{v}^{j,bs}\times] & [{}_{bs}\boldsymbol{e}^j\times] \end{bmatrix} = (\boldsymbol{P}_\Delta + \boldsymbol{M}_\Delta)\left({}_{bs}\boldsymbol{J}^{j,bs}\right). \tag{3.31}$$

The step from (3.28) to (3.29) uses the following identities:

$$\boldsymbol{p}^{j,ee} = \boldsymbol{p}^{j,bs} + \boldsymbol{p}^{bs,ee},$$

and Jacobi's identity

$$0 = (a \times b) \times c + (b \times c) \times a + (c \times a) \times b,$$

with $a = \boldsymbol{e}^j, b = \boldsymbol{p}^{bs,ee}$ and $c = {}^{ee}_{bs}\boldsymbol{R}$. $\boldsymbol{S}_\Delta\left({}_{bs}\boldsymbol{J}^{j,bs}\right)$ is the infinitesimal screw transformation corresponding to the infinitesimal displacement twist ${}_{bs}\boldsymbol{J}^{j,bs}$. It has the following interesting properties:

$$\boldsymbol{S}_\Delta(\mathsf{t})\ \mathsf{t} = \boldsymbol{S}_\Delta\left(\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} [\boldsymbol{\omega}\times] & \boldsymbol{O}_{3\times 3} \\ [\boldsymbol{v}\times] & [\boldsymbol{\omega}\times] \end{bmatrix}\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = 0, \tag{3.32}$$

$${}^{ee}_{bs}\boldsymbol{P}\ \boldsymbol{S}_\Delta({}_{ee}\mathsf{t})\ {}^{bs}_{ee}\boldsymbol{P} = \boldsymbol{S}_\Delta({}_{bs}\mathsf{t}), \tag{3.33}$$

$$\boldsymbol{S}_\Delta\left({}_{bs}\boldsymbol{J}^{j,bs}\right) {}^{ee}_{bs}\boldsymbol{S} = {}^{ee}_{bs}\boldsymbol{S}\ \boldsymbol{S}_\Delta\left({}_{ee}\boldsymbol{J}^{j,ee}\right). \tag{3.34}$$

The identity ${}^{ee}_{bs}\boldsymbol{S}\ {}^{bs}_{ee}\boldsymbol{S} = \boldsymbol{I}$ leads to

$$\frac{\partial {}^{bs}_{ee}\boldsymbol{S}}{\partial q^j} = -{}^{bs}_{ee}\boldsymbol{S}\ \boldsymbol{S}_\Delta\left({}_{bs}\boldsymbol{J}^{j,bs}\right) \overset{(3.34)}{=} -\boldsymbol{S}_\Delta\left({}_{ee}\boldsymbol{J}^{j,ee}\right)\ {}^{bs}_{ee}\boldsymbol{S}. \tag{3.35}$$

$\frac{\partial {}^a_b\boldsymbol{P}}{\partial q^j} = \boldsymbol{P}_\Delta \left({}_b\boldsymbol{J}^j \right) {}^a_b\boldsymbol{P}$	$\frac{\partial \boldsymbol{M}^b_a}{\partial q^j} = \boldsymbol{M}_\Delta \left(\boldsymbol{J}^{j,a} \right)$	$\frac{\partial {}^a_b\boldsymbol{S}}{\partial q^j} = \boldsymbol{S}_\Delta \left({}_b\boldsymbol{J}^{j,b} \right) {}^a_b\boldsymbol{S}$

TABLE 3.3 : *Transformation matrix derivatives: summary.*

3.4.6 Partial derivative for twist Jacobian

It is now easy to find a symbolic expression for the derivatives of the Jacobians. For the *inertial representation*, with $j < i$

$$\begin{aligned}
\frac{\partial\ {}_{bs}\boldsymbol{J}^{i,bs}}{\partial q^j} &\overset{\text{Table 3.1}}{=} \frac{\partial}{\partial q^j}\left({}^{ee}_{bs}\boldsymbol{S}\ {}_{ee}\boldsymbol{J}^{i,ee} \right) \\
&= \frac{\partial\ {}^{ee}_{bs}\boldsymbol{S}}{\partial q^j}\ {}_{ee}\boldsymbol{J}^{i,ee} + {}^{ee}_{bs}\boldsymbol{S}\ \frac{\partial\ {}_{ee}\boldsymbol{J}^{i,ee}}{\partial q^j} \\
&\overset{\text{Table 3.2}}{=} \frac{\partial\ {}^{ee}_{bs}\boldsymbol{S}}{\partial q^j}\ {}_{ee}\boldsymbol{J}^{i,ee} \\
&\overset{(3.30)}{=} \boldsymbol{S}_\Delta \left({}_{bs}\boldsymbol{J}^{j,bs} \right) {}^{ee}_{bs}\boldsymbol{S}\ {}_{ee}\boldsymbol{J}^{i,ee} \\
&\overset{\text{Table 3.1}}{=} \boldsymbol{S}_\Delta \left({}_{bs}\boldsymbol{J}^{j,bs} \right)\ {}_{bs}\boldsymbol{J}^{i,bs}.
\end{aligned} \tag{3.36}$$

The second term of the second line vanishes since $j < i$, see Table 3.2. A similar reasoning applies to the *body-fixed* representation ${}_{ee}\boldsymbol{J}^{i,ee}$, with $j > i$

$$\frac{\partial \left({}_{ee}\boldsymbol{J}^{i,ee} \right)}{\partial q^j} = -\boldsymbol{S}_\Delta \left({}_{ee}\boldsymbol{J}^{j,ee} \right)\ {}_{ee}\boldsymbol{J}^{i,ee} \quad (j > i). \tag{3.37}$$

The derivatives for the *hybrid representation* ${}_{bs}\boldsymbol{J}^{i,ee}$, with $j > i$, are calculated as

$$\begin{aligned}
\frac{\partial \left({}_{bs}\boldsymbol{J}^{i,ee} \right)}{\partial q^j} &\overset{\text{Table 3.2}}{=} \frac{\partial\ {}^{ee}_{bs}\boldsymbol{P}}{\partial q^j}\ {}_{ee}\boldsymbol{J}^{i,ee} + {}^{ee}_{bs}\boldsymbol{P}\ \frac{\partial\ {}_{ee}\boldsymbol{J}^{i,ee}}{\partial q^j} \\
&\overset{(3.21),(3.37)}{=} \boldsymbol{P}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee} \right)\ {}^{ee}_{bs}\boldsymbol{P}\ {}_{ee}\boldsymbol{J}^{i,ee} - {}^{ee}_{bs}\boldsymbol{P}\ \boldsymbol{S}_\Delta \left({}_{ee}\boldsymbol{J}^{j,ee} \right)\ {}_{ee}\boldsymbol{J}^{i,ee} \\
&\overset{\text{Table 3.1}}{=} \left\{ \boldsymbol{P}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee} \right) - {}^{ee}_{bs}\boldsymbol{P}\ \boldsymbol{S}_\Delta \left({}_{ee}\boldsymbol{J}^{j,ee} \right)\ {}^{bs}_{ee}\boldsymbol{P} \right\}\ {}_{bs}\boldsymbol{J}^{i,ee} \\
&\overset{(3.33)}{=} \left\{ \boldsymbol{P}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee} \right) - \boldsymbol{S}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee} \right) \right\}\ {}_{bs}\boldsymbol{J}^{i,ee} \\
&\overset{(3.31)}{=} -\boldsymbol{M}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee} \right)\ {}_{bs}\boldsymbol{J}^{i,ee}.
\end{aligned} \tag{3.38}$$

An alternative calculation yields the same result (i.e., for $j > i$):

$$\frac{\partial \left({}_{bs}\boldsymbol{J}^{i,ee} \right)}{\partial q^j} \overset{\text{Table 3.2}}{=} \frac{\partial\ \boldsymbol{M}^{ee}_{bs}}{\partial q^j}\ {}_{bs}\boldsymbol{J}^{i,bs} + \boldsymbol{M}^{ee}_{bs} \frac{\partial \left({}_{bs}\boldsymbol{J}^{i,bs} \right)}{\partial q^j}$$

	inertial	body-fixed	hybrid
	$\frac{\partial\ {}_{bs}\boldsymbol{J}^{i,bs}}{\partial q^j}$	$\frac{\partial\ {}_{ee}\boldsymbol{J}^{i,ee}}{\partial q^j}$	$\frac{\partial\ {}_{bs}\boldsymbol{J}^{i,ee}}{\partial q^j}$
$j = i$	0	0	$\boldsymbol{P}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee}\right)\ {}_{bs}\boldsymbol{J}^{i,ee}$
$j > i$	0	$-\boldsymbol{S}_\Delta \left({}_{ee}\boldsymbol{J}^{j,ee}\right)\ {}_{ee}\boldsymbol{J}^{i,ee}$	$-\boldsymbol{M}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee}\right)\ {}_{bs}\boldsymbol{J}^{i,ee}$
$j < i$	$\boldsymbol{S}_\Delta \left({}_{bs}\boldsymbol{J}^{j,bs}\right)\ {}_{bs}\boldsymbol{J}^{i,bs}$	0	$\boldsymbol{P}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee}\right)\ {}_{bs}\boldsymbol{J}^{i,ee}$
j n.c. i	0	0	0

TABLE 3.4 : *Joint angle derivatives of a Jacobian's column: summary.*

$$
\begin{aligned}
&\overset{\text{Table 3.2}}{=} && \frac{\partial\ \boldsymbol{M}_{bs}^{ee}}{\partial q^j}\ {}_{bs}\boldsymbol{J}^{i,bs} \\
&\overset{(3.25),(3.27)}{=} && -\boldsymbol{M}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee}\right)\ \boldsymbol{M}_{ee}^{bs}\ \boldsymbol{M}_{bs}^{ee}\ {}_{bs}\boldsymbol{J}^{i,bs} \\
&\overset{\text{Table 3.1}}{=} && -\boldsymbol{M}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee}\right)\ \boldsymbol{M}_{ee}^{bs}\ {}_{bs}\boldsymbol{J}^{i,ee} \\
&\overset{(3.26)}{=} && -\boldsymbol{M}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee}\right)\ {}_{bs}\boldsymbol{J}^{i,ee}.
\end{aligned}
$$

The calculations above are straightforwardly repeated for the case $j < i$:

$$\frac{\partial \left({}_{bs}\boldsymbol{J}^{i,ee}\right)}{\partial q^j} = \boldsymbol{P}_\Delta \left({}_{bs}\boldsymbol{J}^{j,ee}\right)\ {}_{bs}\boldsymbol{J}^{i,ee}. \tag{3.39}$$

Remark that Eqs. (3.38) and (3.39) give the same result for $j = i$. With (3.13), (3.22) and (3.25), both reduce to

$$\frac{\partial \left({}_{bs}\boldsymbol{J}^{i,ee}\right)}{\partial q^i} = \begin{bmatrix} \mathbf{o} \\ {}_{bs}\left[\boldsymbol{e}^i \times (\boldsymbol{e}^i \times \boldsymbol{p}^{i,ee})\right] \end{bmatrix}. \tag{3.40}$$

Tables 3.3 and 3.4 summarize the results of this Section. This Section also presents an alternative way to Eq. (3.14) for the calculation of the Jacobian's time derivative $\dot{\boldsymbol{J}}$. $\dot{\boldsymbol{J}}^i$, the ith column of $\dot{\boldsymbol{J}}$, can be written as

$$\dot{\boldsymbol{J}}^i = \sum_{j=1}^{n} \frac{\partial \boldsymbol{J}^i}{\partial q^j} \frac{\partial q^j}{\partial t} = \sum_{j=1}^{n} \frac{\partial \boldsymbol{J}^i}{\partial q^j} \dot{q}^j. \tag{3.41}$$

So, to calculate the time derivative of the Jacobian in any representation, it suffices to know the joint velocities and the partial derivatives of the Jacobian's columns (in the corresponding representation) with respect to the joint angles.

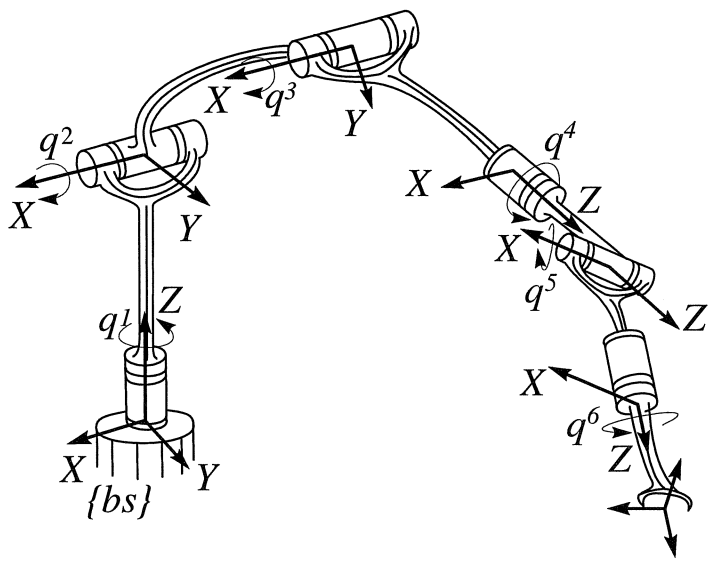


FIGURE 3.4 : *6R kinematic chain.* If all six joint angles are zero, the manipulator is in fully upward position, with all reference frames parallel.

3.4.7 Prismatic joints

Similar to the calculation of the Jacobian matrix $\boldsymbol{J}$ itself, replacing a revolute joint by a prismatic joint simplifies the calculation of the derivative $\partial \boldsymbol{J}^i / \partial q^j$. Indeed, if joint i is prismatic, Eqs. (3.12) and (3.13) simplify to

$$\boldsymbol{J}^{i,bs} = \boldsymbol{J}^{i,ee} = \begin{bmatrix} \mathbf{o} \\ \boldsymbol{e}^i \end{bmatrix}. \tag{3.42}$$

It is then straighforward to see that $\boldsymbol{P}_\Delta$ and $\boldsymbol{S}_\Delta$ get some more zeros, but formally the relationships listed in Table 3.3 do not change. Hence also the symbolic expressions for the Jacobian derivatives do not change.

3.5 Numerical results

This Section applies Eq. (3.36), for the joint angle derivative in the inertial representation, to the example of the 6R kinematic chain whose position and velocity kinematics are described by Featherstone (1983), see also FIG. 3.4. All joint angles equal to 1.0 rad result

in the Jacobian

$$\boldsymbol{J} = \begin{bmatrix} 0 & -0.54 & -0.54 & -0.77 & -0.59 & -0.64 \\ 0 & -0.84 & -0.84 & 0.49 & -0.27 & -0.43 \\ 1 & 0 & 0 & -0.42 & 0.77 & -0.64 \\ 0 & 841.47 & 1068.80 & -718.62 & 617.28 & 206.94 \\ 0 & -540.30 & -686.27 & -1119.18 & -142.34 & -1124.12 \\ 0 & 0 & 420.74 & 0 & 423.84 & 555.45 \end{bmatrix}. \tag{3.43}$$

In this calculation, the length of the first link is 1000 mm, the second link measures 500 mm, and the third one 400 mm. Consider the derivative with respect to the third joint angle. Then, from (3.31),

$$\boldsymbol{S}_{\Delta}\left(\boldsymbol{J}^3\right) = \begin{bmatrix} 0 & 0 & -0.84 & 0 & 0 & 0 \\ 0 & 0 & 0.54 & 0 & 0 & 0 \\ 0.84 & -0.54 & 0 & 0 & 0 & 0 \\ 0 & -420.74 & -686.27 & 0 & 0 & -0.84 \\ 420.74 & 0 & -1068.80 & 0 & 0 & 0.54 \\ 686.27 & 1068.80 & 0 & 0.84 & -0.54 & 0 \end{bmatrix}, \tag{3.44}$$

and, from (3.36),

$$\frac{\partial \boldsymbol{J}}{\partial q^3} = \boldsymbol{S}_{\Delta}\left(\boldsymbol{J}^3\right)\boldsymbol{J} = \begin{bmatrix} 0 & 0 & 0 & 0.35 & -0.64 & 0.54 \\ 0 & 0 & 0 & -0.22 & 0.41 & -0.34 \\ 0 & 0 & 0 & -0.91 & -0.35 & -0.30 \\ 0 & 0 & 0 & 78.88 & -770.06 & 152.63 \\ 0 & 0 & 0 & 122.85 & -835.58 & 714.36 \\ 0 & 0 & 0 & 0 & -89.94 & -117.87 \end{bmatrix}. \tag{3.45}$$

The correctness of the symbolic formulas is illustrated by comparing the following Jacobians:

1. $\boldsymbol{J}^{num}$, which is the numerically calculated Jacobian in the position $\boldsymbol{q} = (1.0\ 1.0\ 1.3\ 1.0\ 1.0\ 1.0)^T$, i.e., $dq^3 = 0.3$ rad. This calculation yields

$$\boldsymbol{J}^{num} = \begin{bmatrix} 0 & -0.54 & -0.54 & -0.63 & -0.76 & -0.47 \\ 0 & -0.84 & -0.84 & 0.40 & -0.15 & -0.54 \\ 1 & 0 & 0 & -0.67 & 0.63 & -0.70 \\ 0 & 841.47 & 1068.80 & -663.21 & 396.05 & 272.21 \\ 0 & -540.30 & -686.27 & -1032.89 & -386.83 & -891.34 \\ 0 & 0 & 420.74 & 0 & 388.49 & 509.12 \end{bmatrix}. \tag{3.46}$$

2. $\boldsymbol{J}^{sym}$, which is the same Jacobian but calculated with the symbolically derived $\partial \boldsymbol{J}/\partial q^3$. $\boldsymbol{J}^{sym}$ is calculated iteratively (because the joint increment is far from infinitesimal!) in n steps:

$$\boldsymbol{J}_0 \;=\; \boldsymbol{J}(dq^3 = 0);$$

$$\begin{aligned} \boldsymbol{J}_i &= \boldsymbol{J}_{i-1} + \frac{\partial \boldsymbol{J}_{i-1}}{\partial q^3}\frac{dq^3}{n}, \quad \text{for } i = 1 \ldots n; \\ \boldsymbol{J}^{sym} &= \boldsymbol{J}_n. \end{aligned}$$

A one step estimate yields

$$\boldsymbol{J}^{sym}(1) = \begin{bmatrix} 0 & -0.54 & -0.54 & -0.66 & -0.78 & -0.48 \\ 0 & -0.84 & -0.84 & 0.42 & -0.14 & -0.54 \\ 1 & 0 & 0 & -0.69 & 0.66 & -0.73 \\ 0 & 841.47 & 1068.80 & -694.96 & 386.26 & 252.73 \\ 0 & -540.30 & -686.27 & -1082.33 & -393.02 & -909.81 \\ 0 & 0 & 420.74 & 0 & 396.86 & 520.09 \end{bmatrix}.$$

$n = 5$ yields

$$\boldsymbol{J}^{sym}(5) = \begin{bmatrix} 0 & -0.54 & -0.54 & -0.63 & -0.77 & -0.47 \\ 0 & -0.84 & -0.84 & 0.41 & -0.15 & -0.54 \\ 1 & 0 & 0 & -0.67 & 0.63 & -0.71 \\ 0 & 841.47 & 1068.80 & -669.30 & 393.02 & 268.21 \\ 0 & -540.30 & -686.27 & -1042.89 & -389.25 & -893.92 \\ 0 & 0 & 420.74 & 0 & 389.99 & 511.09 \end{bmatrix}.$$

And $n = 50$ yields

$$\boldsymbol{J}^{sym}(50) = \begin{bmatrix} 0 & -0.54 & -0.54 & -0.63 & -0.76 & -0.47 \\ 0 & -0.84 & -0.84 & 0.40 & -0.15 & -0.54 \\ 1 & 0 & 0 & -0.67 & 0.63 & -0.70 \\ 0 & 841.47 & 1068.80 & -663.81 & 395.73 & 271.86 \\ 0 & -540.30 & -686.27 & -1033.82 & -387.09 & -891.57 \\ 0 & 0 & 420.74 & 0 & 388.63 & 509.31 \end{bmatrix}.$$

3.6 Conclusions

This Chapter contributes new material for:

1. The definition and transformation properties of the three most common representations for a rigid body twist and for the Jacobian matrix of a serial kinematic chain. These transformations are: the inertial, body-fixed and hybrid representations.

2. The symbolic calculation of the derivatives of the Jacobian matrix for a serial kinematic chain, with respect to any joint angle in the chain, and with respect to time. Only the Jacobian of the chain in its current position is required.

The derivatives are given for all three representations. Numerical results in the inertial representation illustrate the correctness of the symbolically derived formulas.

The material is developed for a general serial kinematic chain with an arbitrary number of revolute joints; with some simplifications it applies to manipulators containing one or more prismatic joints too.

Chapter 4

Kinematic models

4.1 Introduction

The goal of a compliant motion task is to move the *MO* while it maintains contact with the *ENV* in a controlled way. Too large contact forces must be avoided, and contacts should only be broken if explicitly desired by the user. In order to achieve this goal, motion specification, set-point and tracking control modules, FIG. 1.1, must closely cooperate. This thesis develops a mathematical framework in which all three modules can be integrated; this Chapter focusses on the description of the instantaneous and contact maintaining motion freedom of the *MO* under an arbitrary combination of motion constraints. The basic motion constraint model relies on a simple and intuitive kinematic analogy: the *virtual (contact) manipulator*. A virtual manipulator model requires only a minimum number of parameters if these parameters are inspired by the familiar first and second order geometry of the contacting surfaces.

Section 4.2 constructs the virtual manipulator for the generic *surface-surface* contact between two smooth objects. The mathematical representation of the motion degrees-of-freedom of this contact is given in terms of the manipulator's Jacobian matrices. This mathematical model is then used to describe how the nominal motion of the *MO* is specified, and how the model evolves due to this nominal motion. Section 4.3 covers all other contact types —in fact, there aren't that many— as limit cases of the generic *surface-surface* contact. Section 4.4 describes how to model multiple contacts in parallel, and how the motion specification copes with this more complex situation.

4.2 Point contact between smooth surfaces

The most general contact *type*, the *surface-surface* contact, occurs when *MO* and *ENV* both have smooth surfaces in the immediate neighbourhood of the contact, and touch each other in one single point only. If both objects have *ideal* surfaces —i.e., infinitely rigid, no friction— the general point contact allows five independent motion degrees-of-freedom, and generates a one-dimensional space of reaction forces.

It is not difficult to think of the *MO* as being connected to the *ENV* by a five degrees-of-freedom kinematic chain —a *virtual (contact) manipulator*— that gives it the same instantaneous and contact preserving motion freedom as the physical contact, Bruyninckx, De Schutter & Allotta (1991), De Schutter, Bruyninckx & Dutré (1993). The "contact preserving" feature of the model is important: in principle, the *MO* can always break the contact and move away in free space, while a similar motion in the inverse direction is physically not allowed by the contact. Ohwovoriole & Roth (1981) introduced the following terminology: a contact preserving motion is called *reciprocal*, a contact breaking motion is called *repelling*, and its inverse is called *contrary*. In normal circumstances, the user wants his compliant motions to be reciprocal, hence the fact that the repelling and contrary motions are not in the virtual manipulator model is not restrictive.

In general, each contact can be modelled by an infinite number of virtual manipulators. An important design requirement is that the virtual manipulator remains an accurate model of the constraint over an as large as possible range of motion. Of course, a trade-off has to be made between accuracy and practical usability of the virtual manipulator model. A good compromise is to choose the first and second order geometry of the surfaces as the basic building blocks for the virtual manipulator. If the geometric parameters of the model are correct, this model accurately describes the instantaneous velocities and accelerations that the *MO* can use to move compliantly over the *ENV*.

This Section first discusses the relevant geometric parameters, i.e., those of the so-called *principal frame*. Then, it constructs the virtual contact manipulator on top of these parameters, and introduces its mathematical representation, that is, the twist and wrench Jacobians. The twist Jacobian is a model for the nominal motion freedom of the *MO*, and is used to predict the nominal evolution of the virtual manipulator. The wrench Jacobian is a model for all possible ideal reaction forces.

4.2.1 Geometric model: the principal frames

The geometry of the *MO* around the contact point determines an orthogonal reference frame, called the *principal frame*, O'Neill (1966), and denoted $\{pri\}$ in this text, FIG. 4.1. Its Z axis is the inward normal, its X axis is the direction of maximum curvature, and its Y axis the direction of minimum curvature. These maximum and minimum curvature directions are always orthogonal and uniquely defined, unless the surface is discontinuous or locally a sphere or a plane. In the case of a discontinuity, one or both of the minimum and maximum curvatures degenerates; in the case of a sphere or a plane, both curvatures are equal. Similarly, the *ENV* has a principal frame at the contact point too; it is denoted $\{pri'\}$. The Z axes of both principal frames lie on the same line, but they point in the opposite direction. In general, the X axes are rotated with respect to each other about the Z axis.

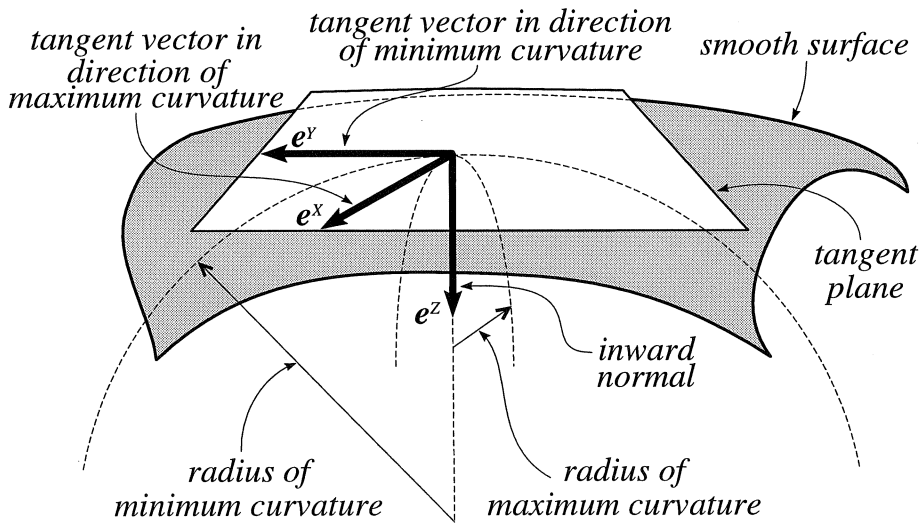


FIGURE 4.1 : *The principal frame of a smooth surface.* It consists of the inward normal unit vector $\boldsymbol{e}^Z$, the tangent unit vector in the direction of maximum curvature, $\boldsymbol{e}^X$, and the tangent unit vector in the direction of minimum curvature, $\boldsymbol{e}^Y$.

4.2.2 Kinematic model: the virtual contact manipulator

The five motion degrees-of-freedom of a *surface-surface* contact are combined into three groups:

1. **Slip**: these two degrees-of-freedom move the *MO* without moving the contact point over the surface of the *ENV*. The slip degrees-of-freedom are kinematically modelled by two revolute joints, R^- and R^+, to which the *MO* is connected as "end effector," FIG. 4.2:

   (a) R^- has its joint axis parallel to the direction of maximum curvature. The joint is located at the centre of minimum curvature, a distance r^- away from the contact point; r^- is the radius of minimum curvature. Applying a joint velocity $\dot{q}^-$ to R^- moves the contact point tangentially in the direction of minimum curvature, i.e., along $\boldsymbol{e}^Y$, with a translational velocity of $\dot{q}^- r^-$.

   (b) R^+ has its joint axis parallel to the direction of minimum curvature. It is located at the centre of maximum curvature, a distance r^+ away from the contact point. A joint velocity $\dot{q}^+$ generates the slipping velocity in the direction of minimum curvature, i.e., along $\boldsymbol{e}^X$, with a translational velocity of $\dot{q}^+ r^+$.

2. **Slide**: these two degrees-of-freedom move the *MO* without moving the contact point over the surface of the *MO*. Their kinematic model is similar to the slip model, but now based on the curvature parameters of the *ENV*. Notationally, the symbols get a prime: R'^-, R'^+, r'^- and r'^+.

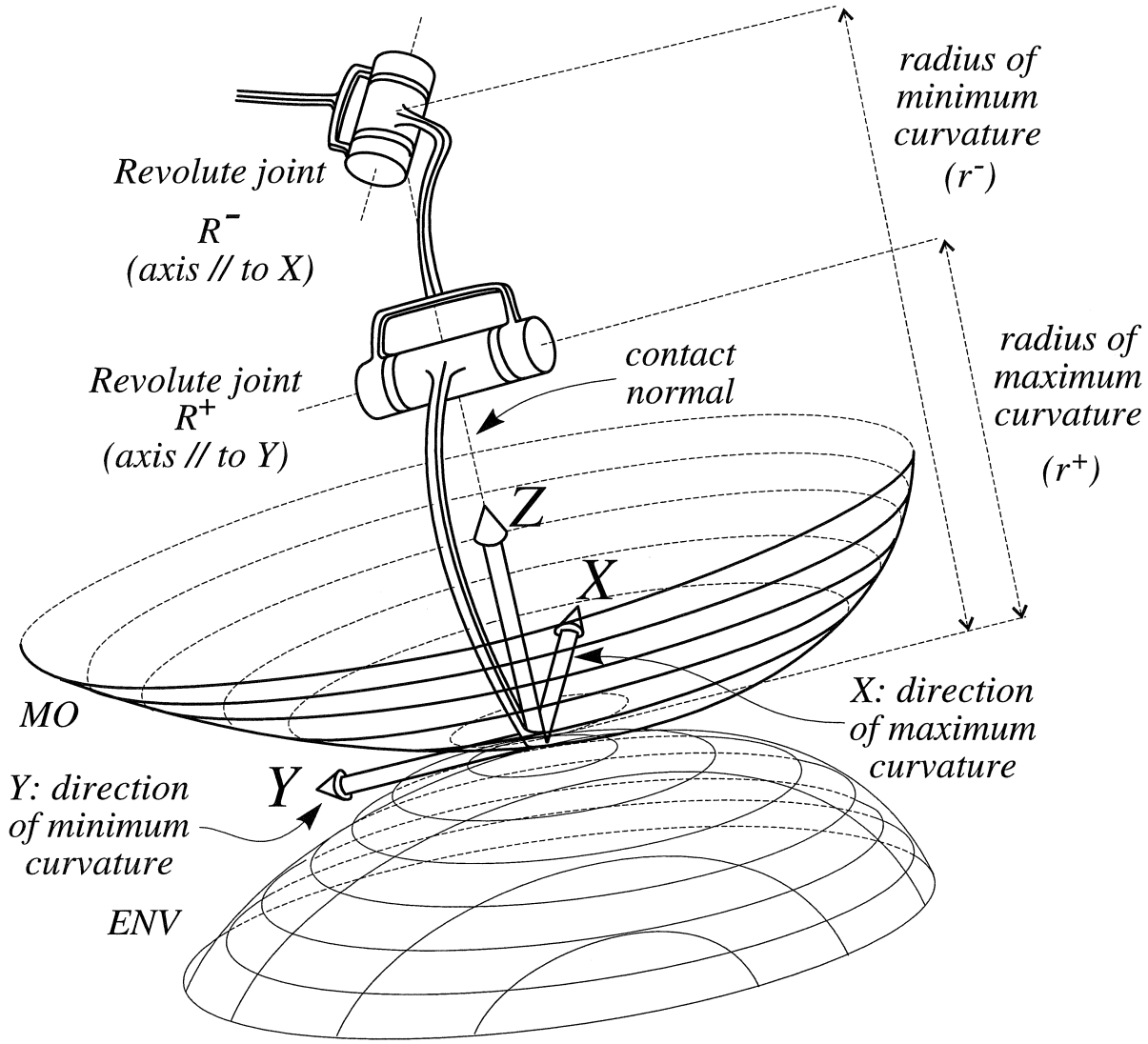


FIGURE 4.2 : *Slip manipulator for a point contact between two smooth surfaces.*

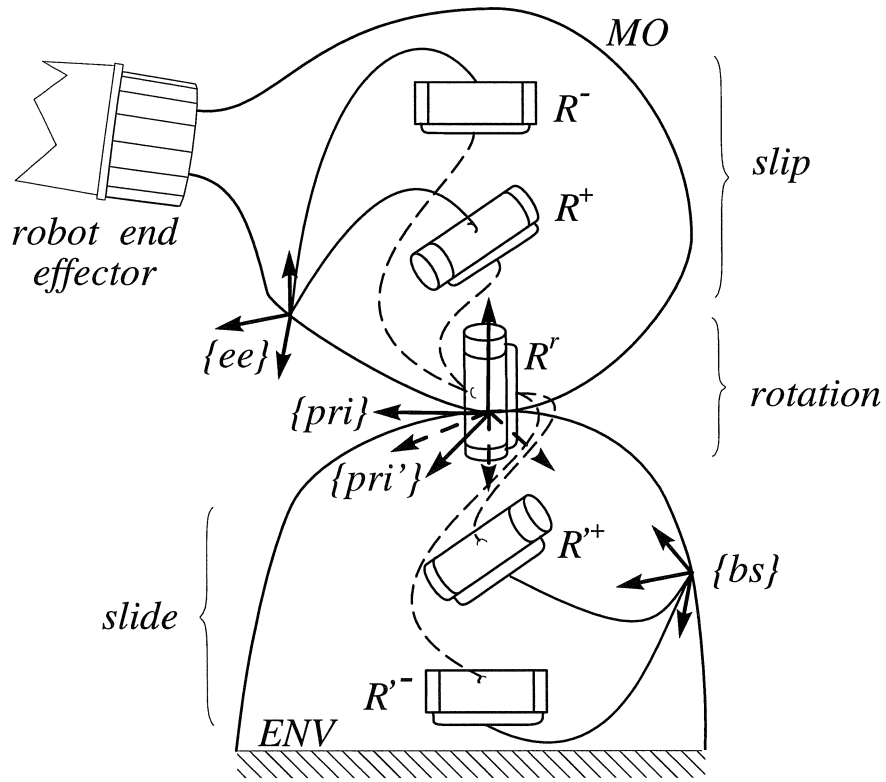


FIGURE 4.3 : *Virtual manipulator for a point contact between two smooth surfaces.* The two slip joints are not connected to each other, only to the *MO*. Their individual contributions to the motion of the *MO* are superimposed. A similar remark goes for the two slide joints.

3. **Rotation**: the fifth degree-of-freedom leaves the contact point fixed with respect to both objects, but it rotates the *MO* around the contact normal. This motion is modelled by a revolute joint R^r along the contact normal.

A radius of curvature has a sign, since it is the projection of the centre of curvature on the Z axis of the corresponding principal frame: the sign of r^+ and r^- is taken with respect to the Z axis of $\{pri\}$; r'^+ and r'^- with respect to the Z axis of $\{pri'\}$. It is possible that $|r^-|<|r^+|$.

Rolling occurs frequently in the context of robotic hands. It is a combination of slip and slide, such that there is no relative translational motion between *MO* and *ENV* at the contact point.

The total virtual contact manipulator, FIG. 4.3, consists of the five slip, rotation and slide joints. It links the *MO* to the *ENV*, with the *MO* as "end effector" and the *ENV*

as "base." The "end effector" and "base" are indicated by reference frames $\{ee\}$ and $\{bs\}$ fixed to the *MO* and the *ENV*, respectively. The virtual manipulator is not a *serially connected* chain —contrary to what FIG. 4.2 might suggest— since both slip joints are not rigidly connected; neither are the slide joints. Each of the slip joints moves the "end effector" without moving any other joint in the virtual manipulator; each of the slide joints moves the "end effector," the rotation joint *and* the slip joints, but does not move the other slide joint. The fact that the virtual manipulator is not a "normal" serial chain is not an objection to represent the instantaneous motion freedom of the *MO*, since it is allowed to simply superimpose velocities and accelerations generated by different "sources."

4.2.3 Mathematical model: the Jacobian matrices

Chapter 3 has discussed the definition and calculation of the Jacobian and its derivatives, as well as how its different mathematical representations are related. The most important properties of the Jacobians in the context of this Chapter are repeated here for easy reference. A virtual contact manipulator's Jacobian matrix $\boldsymbol{J}$ is called the *twist Jacobian* of the virtual manipulator, or its *Jacobian* for short. For a manipulator with n joints, the Jacobian maps n joint velocities $\dot{q}^i, i = 1, \ldots, n$ to a Cartesian end effector twist t:

$$\mathsf{t} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = \boldsymbol{J}\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}^1 & \ldots & \boldsymbol{J}^n \end{bmatrix} \begin{bmatrix} \dot{q}^1 \\ \vdots \\ \dot{q}^n \end{bmatrix}. \tag{4.1}$$

t belongs to the vector space span($\boldsymbol{J}$), also called the *twist system* of the manipulator, i.e., the space of all reciprocal Cartesian velocities. $\boldsymbol{J}^i$ is the ith column of $\boldsymbol{J}$; it represents the end effector twist due to the ith joint velocity only.

The time derivative of Eq. (4.1) gives the relationship between the *MO*'s acceleration and the joint velocities and accelerations:

$$\dot{\mathsf{t}} = \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} + \boldsymbol{J}\ddot{\boldsymbol{q}}. \tag{4.2}$$

The twist Jacobian of a general *surface-surface* contact has five columns: two for the slide degrees-of-freedom, one for the rotation degree-of-freedom, and two for the slip degrees-of-freedom. With respect to the principal frames, the Jacobian columns have a simple form:

$${}_{pri'}\boldsymbol{J}^{slide} = \begin{bmatrix} {}_{pri'}\boldsymbol{J}'^{+} & {}_{pri'}\boldsymbol{J}'^{-} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ -r'^{+} & 0 \\ 0 & r'^{-} \\ 0 & 0 \end{bmatrix}, \tag{4.3}$$

$$ {}_{pri}\boldsymbol{J}^{r} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.4} $$

$$ {}_{pri}\boldsymbol{J}^{slip} = \begin{bmatrix} {}_{pri}\boldsymbol{J}^{+} & {}_{pri}\boldsymbol{J}^{-} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ -r^{+} & 0 \\ 0 & r^{-} \\ 0 & 0 \end{bmatrix}. \tag{4.5} $$

$\boldsymbol{J}'^{+}$ represents a rotation about the joint R'^{+} with a unit angular velocity. The other columns of the Jacobian have similar interpretations. In order to express the Jacobian with respect to another reference frame $\{n\}$ —for example, the robot end effector frame, or a reference frame on the *MO* or the *ENV*— the twist in each column of $\boldsymbol{J}$ is transformed to $\{n\}$ by a *screw transformation* matrix $\boldsymbol{S}$, Eq. (3.9):

$$ \begin{aligned} {}_{n}\boldsymbol{J} &= \begin{bmatrix} {}_{n}\boldsymbol{J}^{slide} & {}_{n}\boldsymbol{J}^{r} & {}_{n}\boldsymbol{J}^{slip} \end{bmatrix} \\ &= \begin{bmatrix} {}^{pri'}_{n}\boldsymbol{S}\ {}_{pri'}\boldsymbol{J}^{slide} & {}^{pri}_{n}\boldsymbol{S}\ {}_{pri}\boldsymbol{J}^{r} & {}^{pri}_{n}\boldsymbol{S}\ {}_{pri}\boldsymbol{J}^{slip} \end{bmatrix}. \end{aligned} \tag{4.6} $$

A general *surface-surface* contact has a one-dimensional space of ideal reaction forces. A basis for this space —the so-called *wrench system*— is given by the *wrench Jacobian* $\boldsymbol{G}$. Mathematically, the reaction wrench along the contact normal represents the same geometric combination of a "free vector" ($\boldsymbol{m}$) and a "line vector" ($\boldsymbol{f}$) as the twist t (composed of the free vector $\boldsymbol{\omega}$ and the line vector $\boldsymbol{v}$) corresponding to a rotation about R^{r}: for example, when expressed with respect to the principal frame $\{pri\}$, a contact force of f Newton is given by the wrench $f[0\ 0\ 1\ 0\ 0\ 0]^{T}$; these six numbers represent also an angular velocity of f rad/sec about R^{r}. Moreover, wrenches transform in exactly the same way as twists:

$$ {}_{a}\mathsf{w} = \begin{bmatrix} {}_{a}\boldsymbol{f} \\ {}_{a}\boldsymbol{m} \end{bmatrix} = {}^{b}_{a}\boldsymbol{S} \begin{bmatrix} {}_{b}\boldsymbol{f} \\ {}_{b}\boldsymbol{m} \end{bmatrix} {}_{b}\mathsf{w}. \tag{4.7} $$

Hence, $\boldsymbol{G}$ is numerically equal to $\boldsymbol{J}^{r}$, with respect to any reference frame, at least in the representations of twists and wrenches chosen in this text; i.e., $\mathsf{t} = [\boldsymbol{\omega}\ \boldsymbol{v}]^{T}$ and $\mathsf{w} = [\boldsymbol{f}\ \boldsymbol{m}]^{T}$.

4.2.4 Motion specification

To specify a reciprocal motion for the *MO* is simple: apply desired motions to all joints in the virtual contact manipulator and the *MO* will start to move over the *ENV*. The specified motion can be of first or second order, i.e., joint velocities $\dot{\boldsymbol{q}}$ and/or joint accelerations $\ddot{\boldsymbol{q}}$. This results in a Cartesian velocity twist t and acceleration $\dot{\mathsf{t}}$ for the *MO*, which are reciprocal up to second order.

Motion specification in the "joint space" of the virtual contact manipulator is easier than direct Cartesian motion specification, since it takes the constraints into account *by construction*, and it decomposes the motion of the MO into slip, slide and rotation components, which have easy and intuitive interpretations. Moreover, it offers the possibility to transform a "first order" Cartesian motion specification into a "second order" Cartesian motion specification: assume, for example, that the desired Cartesian motion is pure slip, with a translational velocity of the instantaneous contact point of v mm/sec, in a direction that makes an angle θ with the direction of maximum curvature of the MO, i.e., the X axis of $\{pri\}$. From Eq. (4.5) one easily finds the corresponding angular velocity set-points for both slip joints R^+ and R^-:

$$\dot{q}^+ = -\frac{\cos(\theta)v}{r^+}, \quad \dot{q}^- = \frac{\sin(\theta)v}{r^-}. \tag{4.8}$$

The resulting Cartesian twist of the MO, with respect to the contact frame is:

$$\mathsf{t} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = \boldsymbol{J}^+\dot{q}^+ + \boldsymbol{J}^-\dot{q}^- = \begin{bmatrix} \dot{q}^- \\ \dot{q}^+ \\ 0 \\ \cos(\theta)v \\ \sin(\theta)v \\ 0 \end{bmatrix}. \tag{4.9}$$

Equation (4.2), with $\ddot{\boldsymbol{q}} = 0$, yields:

$$\dot{\mathsf{t}} = \dot{\boldsymbol{J}}^+\dot{q}^+ + \dot{\boldsymbol{J}}^-\dot{q}^-.$$

The time derivatives of $\boldsymbol{J}^+$ and $\boldsymbol{J}^-$, in the hybrid representation, as defined in Chapter 3, are explicitly calculated in Appendix A, Eqs. (A.15) and (A.16). Together with Eq. (3.41), this gives:

$$\dot{\boldsymbol{J}}^+ = \frac{\partial \boldsymbol{J}^+}{\partial q^+}\dot{q}^+ = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^+\dot{q}^+ \end{bmatrix}, \quad \dot{\boldsymbol{J}}^- = \frac{\partial \boldsymbol{J}^-}{\partial q^-}\dot{q}^- = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^-\dot{q}^- \end{bmatrix}.$$

Hence:

$$\dot{\mathsf{t}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ kv^2 \end{bmatrix}, \quad \text{with } k = \frac{\cos^2\theta}{r^+} + \frac{\sin^2\theta}{r^-}. \tag{4.10}$$

The expression for k is Euler's equation, O'Neill (1966). It gives the normal curvature in a direction θ with respect to the principal X direction. The example above shows that a "first

order" user input —i.e., a direction of motion θ and a desired velocity set-point v— results in a second order motion specification for the *MO*, i.e., desired velocity and acceleration set-points. This is very useful in force controlled compliant motion: the expected change in orientation of the contact normal, as well as the expected normal acceleration, are valuable feedforward information for, respectively, the tracking and force set-point controllers. Hence, a good geometric second order model reduces the tracking and force errors, even without "real" second order motion specification. This is illustrated by the experiments in Chapter 7.

Contact frame. The principal directions of both $\{pri\}$ and $\{pri'\}$ are "object oriented" concepts which are not always relevant for the task to be executed. Hence, a task-specific *contact frame* $\{con\}$ could be introduced between $\{pri\}$ and $\{pri'\}$. Its Z axis coincides with the contact normal —this is also the Z direction of $\{pri\}$ and $\{pri'\}$— and its X and Y axes are chosen in function of the task. For example, the X of $\{con\}$ coincides with the desired direction of motion. The contact frame is an intermediate link frame in the virtual contact manipulator; its Z axis coincides with the joint axis of R^r, and the frame can rotate about it independently of the virtual manipulator joints. The contact frame is not needed to model the motion freedom of the *MO*, but can be introduced to facilitate the motion specification.

4.2.5 Nominal evolution

A good model gives not only good instantaneous motion set-points, but reliably predicts its own *evolution* too, i.e., where do the contact points move to, due to the specified joint motions of the virtual contact manipulator. The kinematic model-based approach makes this calculation easy, since evolution corresponds to following the motion of the principal frames $\{pri\}$ and $\{pri'\}$. These are intermediate link frames in the total virtual manipulator, see FIG. 4.3 and the second row of TABLE 4.1. Hence, the evolution is calculated by integrating the specified joint velocities of the virtual contact manipulator, and subsequent adaptation of the link frame positions and the Jacobians. Of course, the real motion will, in general, not fully correspond to this nominally predicted motion. Hence, the model will become inaccurate, and *on-line identification* of these *uncertainties* is required. Uncertainties and their identification are the topics of the following Chapters.

4.3 Other contact types

The general point contact described in the previous paragraphs occurs between a smooth *MO* and a smooth *ENV*. This is the generic contact case; taking the limit of the curvature radii to infinity or to zero generates surfaces with which other elementary contact types are defined. These limit surfaces are:

Planar face. The *MO* (or the *ENV*) becomes a planar surface —also called a *face*, for short— if both r^- and r^+ (or r'^- and r'^+) go to infinity. An infinite radius of curvature

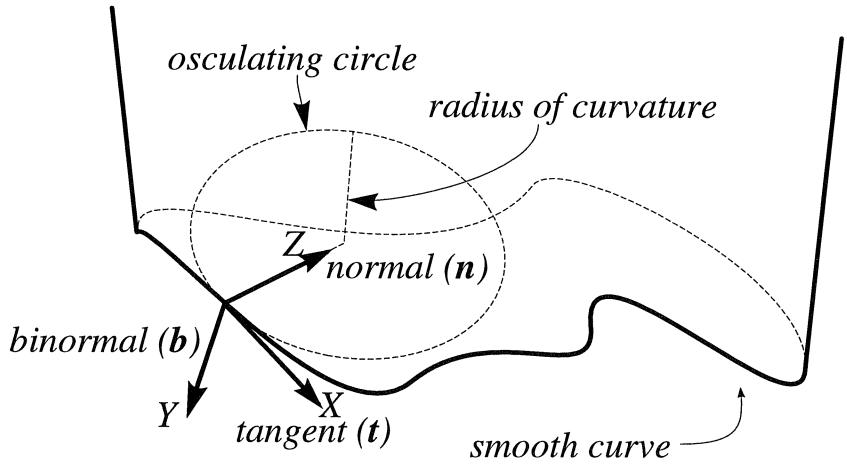


FIGURE 4.4 : *The Frenet frame of a smooth curve.* It consists of the tangent vector $\boldsymbol{t}$, the normal vector $\boldsymbol{n}$ and the bi-normal vector $\boldsymbol{b}$. It is not unambiguously defined if the curve is locally a straight edge.

requires only a simple change in the kinematic model: the corresponding revolute joint is replaced by a prismatic joint. Hence, the virtual contact manipulator —or at least its *slip* part— for a planar face consists of two prismatic joints in the face, along the X and Y directions of the principal frame. These X and Y directions can be chosen arbitrarily.

Smooth curve. If one radius of curvature goes to zero, the surface degenerates to a smooth curve. A zero radius of curvature requires no change in the kinematic model. A smooth curve has no principal frame $\{pri\}$; however, it does have another geometrically determined frame, the *Frenet frame* $\{fre\}$, FIG. 4.4. This frame's X axis is the direction of the tangent to the curve; its Z axis is the normal to the curve, i.e., pointing from the contact point towards the centre of the osculating circle; its Y axis is the bi-normal direction. The kinematic model consists of: a revolute joint R^- along the tangent, and a revolute joint R^+ at the centre of the osculating circle and with its joint axis parallel to the bi-normal.

The contact frame is only partially determined by the Frenet frame, i.e., the contact plane contains the Frenet frame's tangent. The rest of the contact plane's orientation is determined by the other contacting object.

Edge. If one radius of curvature goes to zero, and the other radius of curvature goes to infinity, the surface degenerates to a straight line, also called an edge. The Frenet frame for an edge is not unambiguously defined. The revolute joint R^+ of the curve becomes a prismatic joint P^+ along the edge.

As for a smooth curve, the contact frame's orientation is only partially determined by the geometry of the edge.

contact type: topology of virtual manipulator
surface-surface: $\{ee\} \overbrace{\xrightarrow{R^+}\xrightarrow{R^-}}^{slip} \{pri\} \overbrace{\xrightarrow{R^r} \{con\} \xrightarrow{R^r}}^{rotation} \{pri'\} \overbrace{\xrightarrow{R'^-}\xrightarrow{R'^+}}^{slide} \{bs\}$
surface-curve: $\{ee\} \overbrace{\xrightarrow{R^+}\xrightarrow{R^-}}^{slip} \{pri\} \overbrace{\xrightarrow{R^r} \{con\} \xrightarrow{R^r}}^{rotation} \overbrace{\xrightarrow{R'^-} \{fre'\} \xrightarrow{R'^+}}^{slide} \{bs\}$
face-edge: $\{ee\} \overbrace{\xrightarrow{P^+}\xrightarrow{P^-}}^{slip} \{pri\} \overbrace{\xrightarrow{R^r} \{con\} \xrightarrow{R^r}}^{rotation} \{fre'\} \overbrace{\xrightarrow{R'^-}\xrightarrow{P'^+}}^{slide} \{bs\}$
surface-vertex: $\{ee\} \overbrace{\xrightarrow{R^+}\xrightarrow{R^-}}^{slip} \{pri\} \overbrace{\xrightarrow{R^r} \{con\} \xrightarrow{R^r}}^{rotation} \overbrace{\xrightarrow{R'^-}\xrightarrow{R'^+}}^{slide} \{bs\}$
curve-curve: $\{ee\} \overbrace{\xrightarrow{R^+} \{fre\} \xrightarrow{R^-}}^{slip} \overbrace{\xrightarrow{R^r} \{con\} \xrightarrow{R^r}}^{rotation} \overbrace{\xrightarrow{R'^-} \{fre'\} \xrightarrow{R'^+}}^{slide} \{bs\}$

TABLE 4.1 : *Topology of virtual manipulators for different contact types.* In a contact involving a surface or a face, the contact frame's orientation, i.e., the plane of its X and Y axes, is determined by the surface or the face. In the case of a *curve-curve* contact, both objects determine one direction of the contact plane. A vertex determines only the position of the contact frame's origin, but not its orientation.

Vertex. If both radii of curvature go to zero, the surface degenerates to a vertex. The kinematic model consists of two revolute joints with their axes through the vertex.

The contact frame's orientation is completely determined by the other contacting object, which can only be a surface or a planar face.

FIGURE 4.5 shows the kinematic models of these limit surfaces. The number of different contact types is rather limited, see TABLE 4.2, since not every combination of the surface types above gives a really different contact type, and not all combinations result in stable contacts. TABLE 4.1 lists the kinematic chains for a representative subset of the contact types in TABLE 4.2.

The Jacobians of all contact types follow from Eq. (4.3)-(4.5). However, in some limit cases the rank of the twist Jacobian $\boldsymbol{J}$ decreases to less than five, and the rank of the wrench Jacobian $\boldsymbol{G}$ increases to more than one. For example, in a *face-face* contact all slip and slide revolute joints go to infinity —i.e., they must be replaced by prismatic joints— which implies that the general expressions for the twist Jacobian, Eqs. (4.3) and (4.5), change

into:

$$ {}_{pri'}\boldsymbol{J}^{slide} = {}_{pri}\boldsymbol{J}^{slip} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{4.11} $$

Moreover, these slip and slide Jacobians have become linearly dependent: they both span translational motion in the contact plane. Hence, for a *face-face* contact, slip is indistinguishable from slide. The rank of the total Jacobian reduces to three, i.e., translation in the contact plane (two degrees-of-freedom in the prismatic joints P^+ and P^-) and rotation about the contact normal (one degree-of-freedom in the revolute joint R^r):

$$ {}_{pri}\boldsymbol{J} = \begin{array}{c} \begin{matrix} R^r & P^+ & P^- \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}. \tag{4.12} $$

At the same time, the vector space of ideal reaction forces has grown: not only a force along the contact normal can be generated in the contact, but also moments about any axis in the contact plane. Hence:

$$ {}_{pri}\boldsymbol{G} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.13} $$

Remark The previous paragraphs touched already a bit upon the link with classical differential geometry, as described in text books such as e.g., O'Neill (1966). Appendix C elaborates further on this connection.

4.4 Multiple contacts

Until now, only one single contact has been taken into account. When multiple contacts act on the *MO* in parallel, each individual contact is modelled as in the previous Sections, i.e., by its own virtual manipulator. The construction of this individual virtual manipulator is not influenced by the fact whether or not other contacts occur on the *MO*. However,

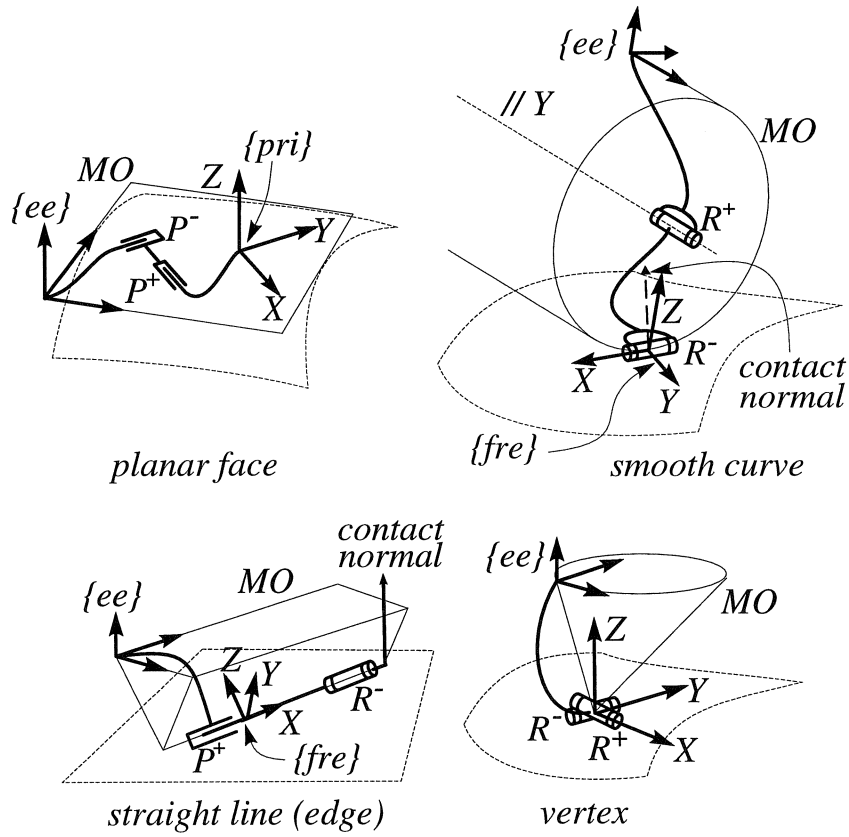


FIGURE 4.5 : *Kinematic models for the different types of contact surfaces.*

Name (*MO-ENV*)	rank($\boldsymbol{J}$)	geometric features
surface-surface	5	
surface-curve	5	$r'^{-} = 0$
surface-vertex	5	$r'^{-} = r'^{+} = 0$
face-face	3	$r^{-} = r^{+} = r'^{-} = r'^{+} = \infty$
curve-surface	5	$r^{-} = 0$
vertex-surface	5	$r^{-} = r^{+} = 0$
curve-curve	5	$r^{-} = r'^{-} = 0$
face-edge	4	$r^{-} = r^{+} = r'^{+} = \infty, r'^{-} = 0$
edge-face	4	$r^{-} = 0, r^{+} = r'^{-} = r'^{+} = \infty$

TABLE 4.2 : *Contact types.*

the twist and wrench spaces for the *MO* do change significantly, and the following subsections describe how. Some other relevant topics for composite motion constraints are also discussed: hyperstaticity, motion specification and redundancy.

4.4.1 Resulting twist and wrench spaces

Assume that n_c contacts act on the *MO* simultaneously. Each of the contacts reduces the *MO*'s motion freedom. The *MO* is able to perform a twist t under the total constraint *if* this twist is allowed by all contacts simultaneously. Mathematically this is described by the following *velocity closure* relations between the twist Jacobians of the virtual contact manipulators :

$$\exists\ \dot{\boldsymbol{q}}^1, \ldots, \dot{\boldsymbol{q}}^{n_c} :\ \mathsf{t} = \boldsymbol{J}^1\dot{\boldsymbol{q}}^1 = \ldots = \boldsymbol{J}^{n_c}\dot{\boldsymbol{q}}^{n_c},$$

or, equivalently, the following homogeneous set of equations must be consistent:

$$\begin{bmatrix} \boldsymbol{J}^1 & -\boldsymbol{J}^2 & 0 & \cdots & 0 \\ \boldsymbol{J}^1 & 0 & -\boldsymbol{J}^3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{J}^1 & 0 & \cdots & 0 & -\boldsymbol{J}^{n_c} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}^1 \\ \dot{\boldsymbol{q}}^2 \\ \vdots \\ \dot{\boldsymbol{q}}^{n_c} \end{bmatrix} = 0. \tag{4.14}$$

Still another way to put this is to say that the twist Jacobian of the total constraint is the *intersection* of the twist Jacobians of the individual contacts:

$$\boldsymbol{J}^{\text{total}} = \bigcap_{i=1}^{n_c} \boldsymbol{J}^i. \tag{4.15}$$

The time derivative of the velocity closure equations gives the *acceleration closure* equations. The Jacobian $\boldsymbol{J}^i, i = 1, \ldots, n_c$ of each contact is as in Eqs. (4.3)-(4.5) or as in one of their limit cases. If the "extended Jacobian matrix" in Eq. (4.14) has n_f independent columns —n_f is always smaller than six— the *MO* is left over with n_f degrees-of-freedom.

While the motion freedom space of the *MO* decreases when more constraints are added, the reaction force space grows. It is spanned by the *concatenation* of the wrench Jacobians of all contacts:

$$\text{span}\left(\boldsymbol{G}^{\text{total}}\right) = \text{span}\left(\left[\boldsymbol{G}^1 \ \ldots \ \boldsymbol{G}^{n_c}\right]\right). \tag{4.16}$$

Equivalently, one can say that the wrench Jacobian of the total constraint is the *vector sum* of the wrench Jacobians of the individual contacts:

$$\boldsymbol{G}^{\text{total}} = \bigoplus_{i=1}^{n_c} \boldsymbol{G}^i. \tag{4.17}$$

If the rank of $\boldsymbol{J}^{\text{total}}$ is n_f, then the rank of $\boldsymbol{G}^{\text{total}}$ is $6 - n_f$.

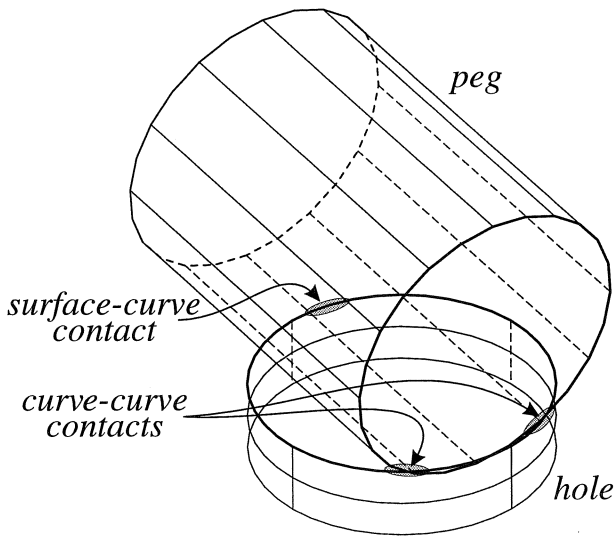


FIGURE 4.6 : *Peg-on-hole contact situation.* If the axes of a cylindrical peg and a compatible hole are not well aligned, the peg rests on the hole at three contact points, depicted in the drawing by shaded ellipses.

4.4.2 Hyperstatic constraint

If contact "1" eliminates n^1 degrees-of-freedom, and contact "2" eliminates n^2 degrees-of-freedom, then the *MO*'s motion freedom in total is reduced by *at most* n^1+n^2. If fewer than n^1+n^2 degrees-of-freedom are eliminated, the constraint is called *hyperstatic*. Reasoning on the reaction wrench space gives a dual look at hyperstaticity: a constraint is hyperstatic if the ideal reaction wrenches of all contacts are not linearly independent. This means that $\text{rank}(\boldsymbol{G}^{\text{total}}) < \sum_{i=1}^{n_c} \text{rank}(\boldsymbol{G}^i)$. An example of a hyperstatic constraint is following a 3D seam, with two *edge-face* contacts as depicted in FIG. 1.10: the moment about the task frame's Z axis has contributions from both *edge-face* contacts. As a consequence, it is impossible to decompose *unambiguously* the measured Z moment into contributions from both contacts.

4.4.3 Motion specification

For one single contact, a possible motion for the *MO* is specified by applying desired joint velocities and/or accelerations to all joints in the corresponding virtual manipulator. For multiple contacts, however, not all joints can be driven independently. The user must choose a set of independent *driving joints*, and specify their desired velocities or accelerations. The motion of the other *passive joints* follows from solving the velocity closure equations (4.14). If n_f^i is the number of joints in the virtual manipulator i, then these equations have $\sum_{i=1}^{n_c} n_f^i$ "unknowns," i.e., the joint velocities of all n_c virtual manipulators. The choice of driving joints can be made off-line, and remains valid as long as the contact

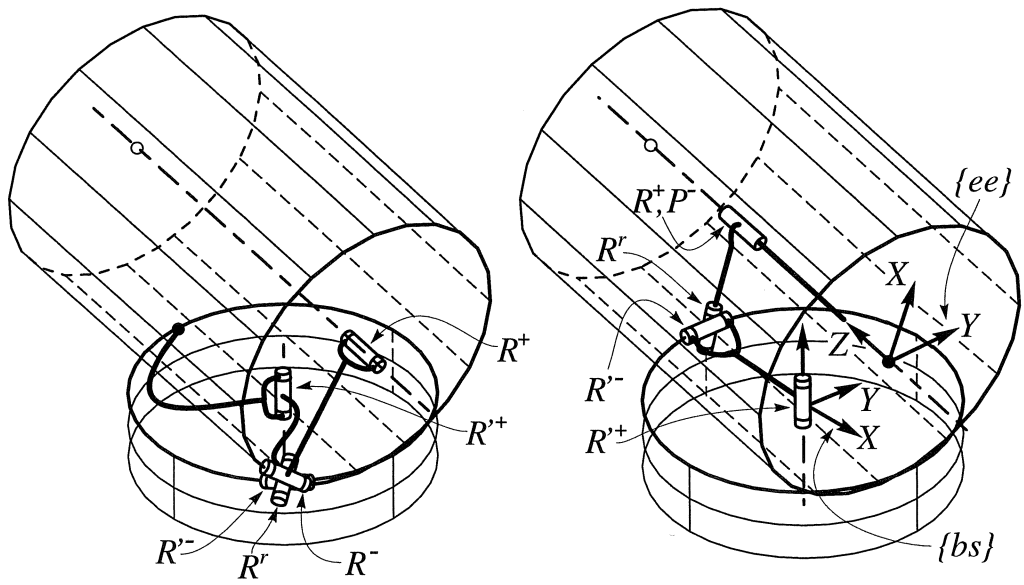


FIGURE 4.7 : *Peg-on-hole: kinematic models.* Left: *curve-curve* contact; right: *surface-curve* contact. All joints are revolute, except the prismatic joint "P^-" along the peg's axis.

situation type does not change and the kinematic model does not become singular.

The *peg-on-hole* contact situation, FIG. 4.6, is a good example to illustrate nominal motion specification under multiple contacts, Bruyninckx, Dutré & De Schutter (1995). The peg is pushed against the hole, and since the axes of peg and hole are not well aligned, the peg cannot be inserted and three contacts occur between peg and hole: one on its outer surface —this contact is of the *surface-curve* type— and two on its bottom rim —both are of the *curve-curve* contact type. FIGURE 4.7 shows the kinematic models for these contacts. In all, the kinematic model has fifteen joints, but since each contact removes one degree-of-freedom from the peg's motion, only three joints can be used as independent drivers. It is obvious to choose such a set of three driving joints as: 1) a rotation about the peg's axis; 2) a rotation about the hole's axis; and 3) a rotation about the revolute joint R'^- —called the *alignment joint*— tangent to the hole's rim at the *surface-curve* contact. Only the alignment joint can reduce the alignment error between peg and hole. Hence, if the purpose of the motion is to align peg and hole, the two other drivers are given a zero-velocity motion specification.

The velocity closure equations, Eq. (4.14), for the peg-on-hole task involve twelve equations and fifteen joint velocities:

$$\begin{bmatrix} \boldsymbol{J}^{sc} & -\boldsymbol{J}^{cc1} & 0 \\ \boldsymbol{J}^{sc} & 0 & \boldsymbol{J}^{cc2} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}^{sc} \\ \dot{\boldsymbol{q}}^{cc1} \\ \dot{\boldsymbol{q}}^{cc2} \end{bmatrix} = 0, \tag{4.18}$$

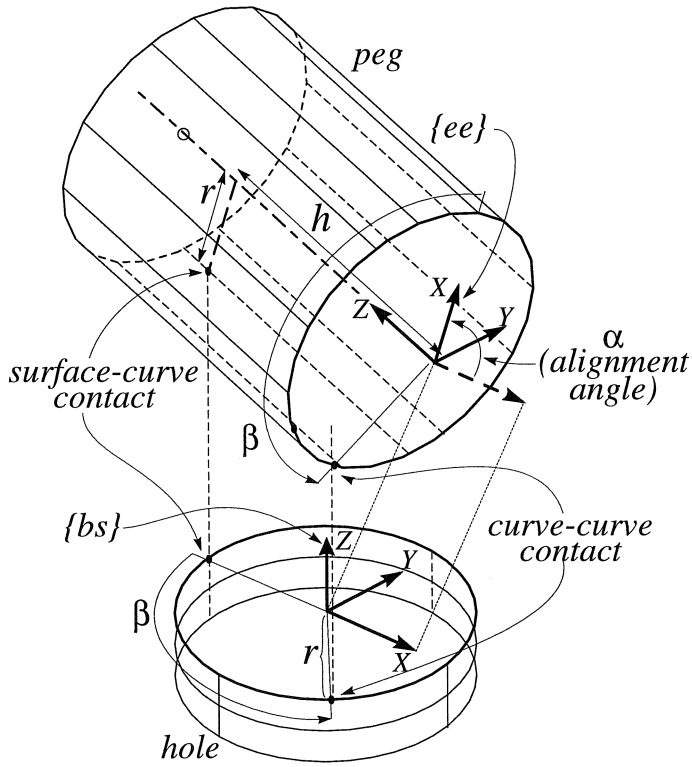


FIGURE 4.8 : *Peg-on-hole: geometric definitions.*

or:

$$\begin{aligned} \mathrm{t}^{MO} &= \boldsymbol{J}^{sc}\,\dot{\boldsymbol{q}}^{sc} \\ &= \boldsymbol{J}^{cc1}\,\dot{\boldsymbol{q}}^{cc1} \\ &= \boldsymbol{J}^{cc2}\,\dot{\boldsymbol{q}}^{cc2}. \end{aligned}$$

$\boldsymbol{J}^{sc}$ is the twist Jacobian of the virtual contact manipulator for the *surface-curve* contact; $\boldsymbol{J}^{cc1}$ and $\boldsymbol{J}^{cc2}$ are the twist Jacobians for the two *curve-curve* contacts. t^{MO} is the twist of the MO resulting from the virtual manipulators' joint velocities. From the symmetry of the contact situation, this set of twelve equations in fifteen unknown joint velocities can be reduced to a 6×6 set. The following paragraphs describe these symmetries; Appendix D contains the mathematical details of the reduction procedure.

Reduction of the number of unknown joint velocities.

1. The virtual manipulators at both *curve-curve* contacts are each other's mirror images with respect to the plane through the centre lines of peg and hole. This means that the corresponding joint positions and velocities are connected by simple relations, i.e., they are equal or opposite depending on the chosen positive rotation directions. Result: five unknown joint velocities less.

2. This Section considers pure alignment only. Hence, the R'^{+}, R^{+} and R^{r} joints of the *surface-curve* contact do not rotate. Result: three unknown joint velocities less.

3. The joint angle of R'^{-} in the *surface-curve* contact is chosen equal to α. Hence, its joint velocity equals the driving velocity $\dot{\alpha}$. Result: one unknown joint velocity less.

In total, only six independent joint velocities remain: the translational velocity $\dot{h}$ of the prismatic joint in the *surface-curve* virtual manipulator, and the joint velocities $\dot{q}^{+}, \dot{q}'^{+}, \dot{q}^{-}, \dot{q}'^{-}$ and $\dot{q}^{r}$ in one of the *curve-curve* virtual contact manipulators.

Reduction of the number of equations. Four equations —two from each row in the velocity closure equations (4.18)— are identically satisfied because there is no rotational velocity about the X and Z axes of the $\{bs\}$ frame in FIG. 4.7. Similarly, two more can be deleted because there is no translational velocity along Y. Finally, the two equations for ω_y must give the known driving velocity $\dot{\alpha}$.

Appendix D solves the remaining 6×6 set of equations analytically, with the velocity of the alignment joint, $\dot{\alpha}$, as driving input:

$$\dot{h} = (c_\beta - 1)\dot{\alpha}, \tag{4.19}$$

$$\dot{q}'^{+} = \dot{q}^{+} = \frac{s_\alpha s_\beta}{2c_\alpha}\dot{\alpha} \tag{4.20}$$

$$\dot{q}'^{-} = \dot{q}^{-} = -\frac{\dot{\alpha}}{2}, \tag{4.21}$$

$$\dot{q}^{r} = \frac{s_\alpha \dot{\alpha}}{2c_\alpha(c_\alpha - 1)}. \tag{4.22}$$

$c_\alpha, s_\alpha, c_\beta$ and s_β are the cosines and sines of the angles α and β. These equations reveal two more symmetries that stem from the less apparent observation that the contact points cut off equal arcs from the bottom rim of the peg and the top rim of the hole; these arcs are spanned by the angles β in FIG. 4.8. Hence, specifying an alignment motion goes as follows:

1. Solve the *position closure* equations to compute the initial alignment error α. The details are described in Appendix D.

2. Choose an alignment velocity profile $\dot{\alpha}(t)$.

3. Apply this velocity to the kinematic chain. Then, Eqs. (4.19)-(4.22), together with the symmetry considerations, give all joint velocities in the kinematic chain.

4. The nominal evolution of the kinematic chain is straightforwardly calculated by integrating these joint velocities.

FIGURE 4.9 shows the nominal motion profile of the peg, with a unit driving velocity $\dot{\alpha}$. The end effector velocity for the manipulating robot is straightforwardly derived from the peg's velocity since the peg is rigidly connected to the robot end effector.

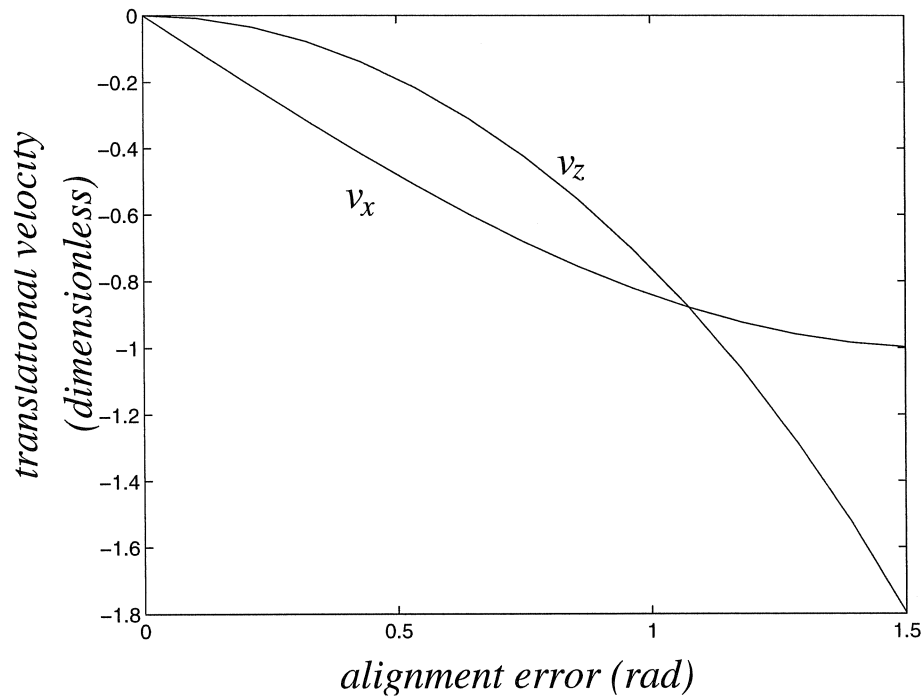


FIGURE 4.9 : *Peg-on-hole: nominal velocity profile.* 'v_x' and 'v_z' are translational velocities of the origin of $\{ee\}$, resulting from a driving angular alignment velocity $\dot{\alpha} = -1 rad/sec$. The translational velocities are non-linear functions of the alignment error α. Peg and hole have unit radius. These dimensionless velocity profiles scale linearly with this radius, and with $\dot{\alpha}$.

4.4.4 Underdetermined motion specification

Even if the rank of the extended twist Jacobian is n_f, the user is allowed to specify only $n_d < n_f$ driving joints. The remaining $n_f - n_d$ degrees-of-freedom are then to be generated by an on-line *redundancy resolution* algorithm. Chapter 7 presents an example of such a redundancy resolution algorithm that is compatible with the kinematic modelling approach used for the nominal motion specification. Another example of an "underdetermined" task is the 3D seam following task of FIG. 1.10. In this constraint, the *MO* has three degrees-of-freedom, of which the user specifies only one: the translational velocity along the seam. The two other degrees-of-freedom, i.e., the angular velocities about the contact edges, are generated on-line by the tracking module.

4.5 Conclusions

This Chapter developed a kinematic model —the virtual contact manipulator— for a contact between two smooth surfaces. The placement of the joints of this virtual manipulator

is inspired by the first and second order geometry of the surfaces —the principal frame for a surface and the Frenet frame for a curve— and the corresponding radii of curvature.

From the *surface-surface* contact, other elementary motion constraints are derived as limit cases, i.e., the curvature radii of one or both surfaces going to zero or to infinity. The resulting set of contact types is small enough for exhaustive enumeration and implementation.

The twist Jacobian $\boldsymbol{J}$, and its time derivative $\dot{\boldsymbol{J}}$ are the mathematical representations —up to second order— of the instantaneous motion freedom of the *MO*, i.e., its Cartesian velocity and acceleration. $\boldsymbol{J}$'s dual, the wrench Jacobian $\boldsymbol{G}$, is the mathematical representation of the possible ideal reaction forces that can be generated by the constraint. The Jacobian matrices corresponding to the virtual manipulators of all elementary motion constraint types have been derived. The *MO*'s motion is specified in the joint space of the virtual manipulator. This reduces the complexity of the motion specification and increases the physical interpretation of the motion, by decomposing it into slip, slide and rotation. The specified motion and the geometric model of the virtual manipulator predict the nominal time evolution of the virtual contact manipulators.

In the case of more than one contact on the *MO*, its motion freedom space is described by the velocity and acceleration closure equations of the complex kinematic chain formed by the virtual contact manipulators of all contacts in parallel. These represent the intersection of the twist and twist rate systems of all individual contacts. The space of ideal reaction forces is spanned by the concatenation of the wrench Jacobians of all contacts, i.e., the vector sum of all individual wrench systems. To specify the *MO*'s motion under the parallel action of several contacts, the user defines a set of independent driving joints among all joints in the virtual manipulators.

A kinematic model as described in this Chapter can only perform appropriately if its geometric parameters —such as the position of the contact point, the direction of the contact normal, and the magnitudes and directions of minimum and maximum curvature— are rather accurate. This can in practice not be guaranteed by an off-line model. Hence, modelling and on-line identification of these geometric parameters is indispensable in general. These topics are treated in the next Chapters. The flexibility of the kinematic models in view of the modelling and identification of these geometric uncertainties is one of their major benefits.

Chapter 5

Uncertainties

5.1 Introduction

Motion specification and evolution prediction give good results only if the instantaneous kinematic model is accurate. Making accurate models off-line is costly. Moreover, even the most accurate model must be matched with the on-line sensor data, and this is costly too. A third possibility remains: provide the controller with simple, approximate contact models, as well as with algorithms to continuously adapt the uncertain parameters in the model to the actual geometry of the contact, as identified by the sensors. Remember that this text assumes that the type of the contact is known, and that only the geometric parameters within the given contact type are uncertain. To discriminate between different contact type hypotheses is considered to be a complementary problem, for which this thesis, however, does not try to find solutions.

Geometric uncertainties as considered in this text have many sources. For example:

1. The geometry of the objects is inaccurately modelled.

2. The position of the objects with respect to the manipulating robot is inaccurately known.

3. The manipulating robot's dynamics are inaccurately controlled, and hence the executed motion differs from the specified motion.

4. The objects, the robot, the fixtures, ..., are not completely rigid; hence the position as well as the motion of the MO as measured by the manipulating robot are incorrect.

As a result, the real contact points occur at different positions than in the model, the contact normals are differently oriented and the time evolution of the constraints differs from the modelled prediction. Hence, the on-line control software must have some means to detect when such errors appear and to quantify their influence. This is called *identification* and it is the topic of the next Chapter. This Chapter prepares the stage for identification: it proposes a model to represent small geometric uncertainties, in such a way that they

are easily integrated into the virtual contact manipulator model of the previous Chapter. Within the context of that Chapter, geometric uncertainties show up as follows:

1. The positions of the principal frames $\{pri\}$ and $\{pri'\}$ with respect to the robot end effector are only approximately known.

2. The structure of the slip and slide Jacobians with respect to these principal frames, Eqs. (4.3)-(4.5), is known —because the type of the contact is known— but not the exact values of the curvature radii r^-, r^+, r'^- and r'^+.

The next Section classifies geometric uncertainties in time varying and time invariant uncertainties on the one hand, and first order and second order uncertainties on the other hand.

Section 5.3 explains how to integrate uncertainties into the nominal kinematic model in a natural way. It treats uncertainties as passive degrees-of-freedom in a virtual uncertainty manipulator.

Uncertainties influence the motion freedom model of the *MO*, and hence also the mathematical representation of this motion freedom model, i.e., the Jacobian matrices. Section 5.4 treats the dependence of the twist and wrench Jacobians on the uncertainties by means of Taylor series expansions of these Jacobians in terms of the uncertainty parameters.

Since the principal frames on the *MO* and the *ENV* are not always unambiguously defined, "false" uncertainties can appear: this means that a change in such a false uncertainty changes the mathematical model (i.e., the Jacobian matrices) but not the physical model (i.e., the instantaneous first or second order motion freedom). These false uncertainties are called *representation uncertainties*. Section 5.5 explains how to discern these representation uncertainties from real uncertainties.

Section 5.6 discusses how uncertainties of several simultaneous contacts influence each other, and in which circumstances the total number of uncertainties can be reduced in a systematic way.

5.2 Uncertainty types

All geometric uncertainties treated in this Chapter are fundamentally of the same nature: they represent the inaccurate knowledge of a parameter in the geometry of the kinematic contact model. However, it is useful to classify them in different sets, each requiring —or allowing— different identification approaches.

A first distinction exists between *first order* and *second order* uncertainties. A first order uncertainty describes inaccurate knowledge of the position of the contact point, and the direction of the contact normal. A second order uncertainty reflects inaccurate knowledge of the objects' curvature parameters: the directions of minimum and maximum curvature, and the corresponding curvature radii.

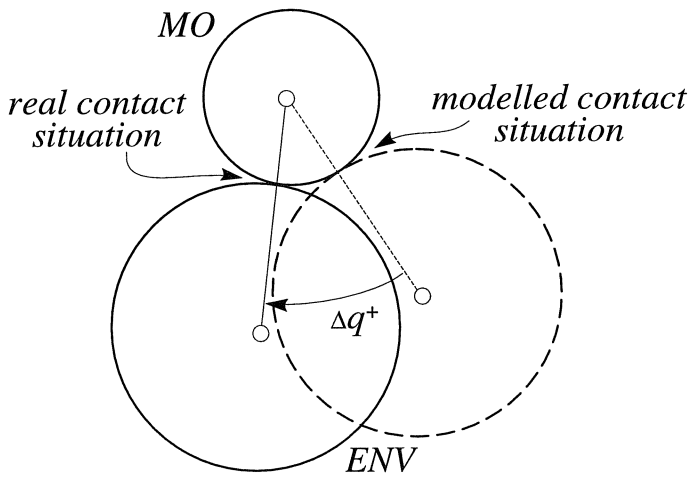


FIGURE 5.1 : *First order time varying uncertainty for a 2D curve-curve contact.* Both *MO* and *ENV* are (modelled as) perfect cylinders with known radius. However, the real position of the contact point differs from the modelled position.

A second distinction exists between *time varying* and *time invariant* uncertainties. Time varying uncertainties continuously change during the motion and require continuous and instantaneous identification. A time invariant uncertainty is an unknown but constant geometric parameter, that stops being an uncertainty from the moment it has been identified. Examples of time varying uncertainties are: 1) the instantaneous radii of maximum and minimum curvature of a smooth but unknown environment; 2) the position of the contact point between smooth curves or surfaces, FIGS. 1.12 and 5.1. An example of a time invariant uncertainty is the position of an object's vertex relative to the robot end effector, FIG. 5.2, if the robot has grasped the object in an inaccurately known way, or if the shape of the *MO* is not exactly known. This relative position error never changes during the task, unless the peg or the gripper breaks or bends. For time invariant uncertainties, identification data gathered over long time intervals can be combined.

In summary, the geometric uncertainties treated in this Chapter are always of one of the following four types: first order time varying, first order time invariant, second order time varying, or second order time invariant. Of course, a general contact situation combines uncertainties of different types at the same time. The following paragraphs define the context within which the following Sections of this Chapter derive identification equations for these different uncertainty types.

First order time varying, FIG. 5.1. At time instant t, the position of the contact point is correct, as well as the twist and wrench Jacobians. This means the first order motion freedom is accurately modelled. At time instant $t + \Delta t$ however, the different uncertainty sources have caused the contact point to drift a bit off its nominally predicted position.

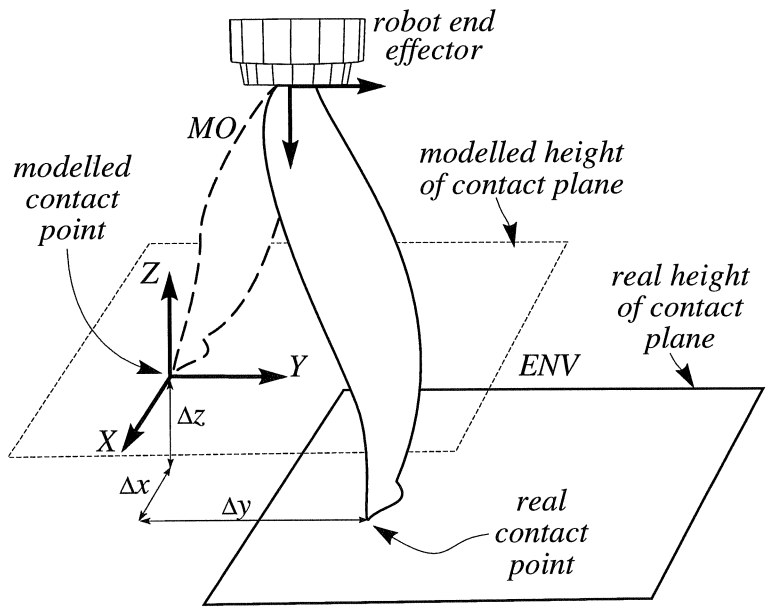


FIGURE 5.2 : *First order time invariant uncertainty for a vertex-face contact.* The *MO* makes a *vertex-face* contact with a horizontal plane. The relative position uncertainty $[\Delta x\ \Delta y\ \Delta z]^T$ between the vertex and the end effector does not change when the *MO* moves.

First order time invariant, FIG. 5.2. The twist and wrench Jacobians with respect to the contact are exactly known, but the relative position of this contact with respect to the manipulating robot is uncertain. This uncertainty does not change over time, and hence the error in the motion specification persists until the uncertainty is identified. From then on it is no uncertainty anymore for the rest of the task.

Second order time varying, FIG. 5.3. The position of the contact point is known at time instant t, as well as the type and contents of the Jacobians (including the curvature parameters). This means that the first and second order motion freedom are accurately modelled. At time instant $t + \Delta t$, however, the curvature parameters have changed a bit, due to the motion of the *MO*.

Second order time invariant, FIG. 5.4. From the start, an error in the curvature parameters exists. This error does not change during the task, and it continuously gives wrong evolution predictions until it is identified.

The following notations are used further on: the model has n_u geometric uncertainty parameters, collected in a so-called *uncertainty parameter set* $\mathcal{U} = \{\upsilon_1, \ldots, \upsilon_{n_u}\}$ (υ is the Greek letter "upsilon," referring to the first letter of the word "uncertainty"); if necessary, extra superscripts "1" and "2" indicate first and second order uncertainty parameters, respectively. No special notation will be needed to discriminate between time varying and time invariant uncertainties.

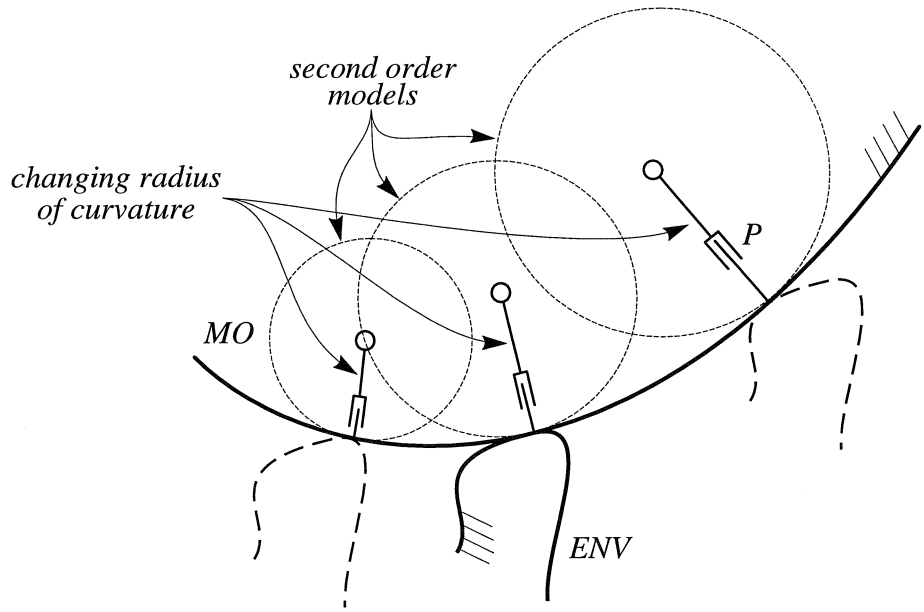


FIGURE 5.3 : *Second order time varying uncertainties for a 2D curve-curve contact.* This drawing depicts three consecutive positions of the *MO* in a relative motion with respect to the *ENV*. The change in the *MO*'s radius of curvature is modelled by a prismatic joint *P*.

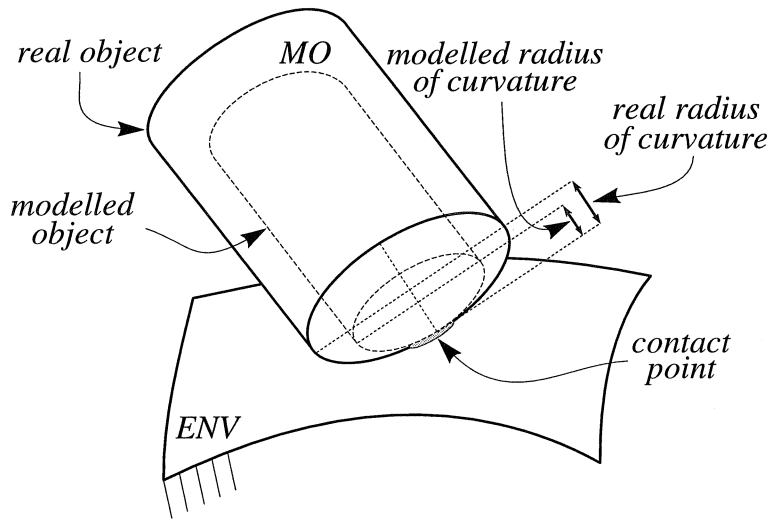


FIGURE 5.4 : *Second order time invariant uncertainty for a curve-surface contact.* In the model, the cylindrical peg has a smaller radius than in reality. This causes an error in the second order motion freedom of the *MO*.

5.3 Kinematic model for uncertainties

This Section treats the general *surface-surface* contact only. The changes required for the other elementary contact types listed in TABLE 4.2 are straightforward. It is assumed that a nominal virtual contact manipulator model exists, i.e., the principal frames are placed somewhere in space, and connected with the slide, rotation and slip joints of the virtual manipulator.

First order uncertainties correspond to errors in the position of the contact point and/or the direction of the contact normal. Time varying and time invariant first order uncertainties are modelled differently:

1. According to the application context sketched in the previous Section, time varying first order uncertainties represent the errors made in predicting the evolution of the contact point over the surfaces of *MO* and *ENV*, as well as the errors caused by a wrong orientation of the contact normal. Modelling these errors requires no extensions to the nominal kinematic model, since they can be modelled by errors in the joint angles of the slip and slide manipulators: i.e., adjusting the current joint angles q^+, q^-, q'^+ and q'^- by small increments $\Delta q^+, \Delta q^-, \Delta q'^+$ and $\Delta q'^-$, moves the contact over the surfaces of both *MO* and *ENV* until it is correctly placed, Bruyninckx & De Schutter (1993). FIGURE 5.1 visualizes this idea on a 2D example.

2. Time invariant first order uncertainties, on the other hand, have a much less precise origin. FIGURE 5.2 depicts a time invariant situation with only three uncertainty parameters because the orientation of the contact plane with respect to the robot end effector is assumed to be known. In general, *six* uncertainty parameters are required. A kinematic model for these is fairly simple: a six degrees-of-freedom manipulator. The nominal kinematic model can supply five of these six degrees-of-freedom.

Second order uncertainties represent uncertainties in the curvature parameters, i.e., the directions of maximum and minimum curvature, and the corresponding curvature radii. If the surfaces of the contacting objects cannot be perfectly modelled by one single virtual contact manipulator, playing with the slip and slide joint angles alone does not suffice to continuously keep the real contact on the modelled surface. A change in the radii and/or directions of curvature must compensate the part of the contact's time evolution that continuously causes first order model errors. Time varying or time invariant second order uncertainties get identical kinematic models: the curvature uncertainties of the *MO* are integrated into the slip manipulator by two *prismatic* joints P^+ and P^- between the contact point and R^+ and R^- respectively, see FIG. 5.3 for a 2D model. Similarly for the *ENV*: the prismatic joints P'^+ and P'^- extend the slide manipulator. The "joint position errors" of these prismatic joints are denoted $\Delta r^-, \Delta r^+, \Delta r'^-$ and $\Delta r'^+$.

5.4 Mathematical model

Uncertainties disturb the nominal Jacobians, since the geometric parameters on which the Jacobians are constructed are wrong. The influence of an uncertainty υ on the Jacobians is given by the derivatives with respect to υ. Formally:

$$\boldsymbol{J}(\mathcal{U}) = \boldsymbol{J}(\mathcal{U}=0) + \sum_{i=1}^{n_u} \upsilon_i \frac{\partial \boldsymbol{J}}{\partial \upsilon_i}, \tag{5.1}$$

$$\dot{\boldsymbol{J}}(\mathcal{U}) = \dot{\boldsymbol{J}}(\mathcal{U}=0) + \sum_{i=1}^{n_u} \upsilon_i \frac{\partial \dot{\boldsymbol{J}}}{\partial \upsilon_i}, \tag{5.2}$$

$$\boldsymbol{G}(\mathcal{U}) = \boldsymbol{G}(\mathcal{U}=0) + \sum_{i=1}^{n_u} \upsilon_i \frac{\partial \boldsymbol{G}}{\partial \upsilon_i}. \tag{5.3}$$

The following paragraphs elaborate this Taylor series expansion a bit more for the example of the twist Jacobian $\boldsymbol{J}$; the same procedure can be repeated for the other Jacobians. Assume a contact situation with n_u^1 first order uncertainty parameters, and n_u^2 second order uncertainty parameters, collected in the uncertainty parameter sets $\mathcal{U}^1 = \{\upsilon_1^1, \ldots, \upsilon_1^{n_u^1}\}$ and $\mathcal{U}^2 = \{\upsilon_1^2, \ldots, \upsilon_2^{n_u^2}\}$. Then $\boldsymbol{J}$'s series expansion in terms of these uncertainty parameters, Eq. (5.1), gives:

$$\boldsymbol{J}(\mathcal{U}^1, \mathcal{U}^2) = \boldsymbol{J}(0,0) + \sum_{j=1}^{n_u^1} \frac{\partial \boldsymbol{J}}{\partial \upsilon_1^j} \upsilon_1^j + \sum_{j=1}^{n_u^2} \frac{\partial \boldsymbol{J}}{\partial \upsilon_2^j} \upsilon_2^j. \tag{5.4}$$

The time derivative of this series expansion yields:

$$\begin{aligned} \dot{\boldsymbol{J}}(\mathcal{U}^1, \mathcal{U}^2) &= \dot{\boldsymbol{J}}(0,0) \\ &+ \sum_{j=1}^{n_u^1} \left\{ \frac{\partial \boldsymbol{J}}{\partial \upsilon_1^j} \dot{\upsilon}_1^j + \frac{\partial \dot{\boldsymbol{J}}}{\partial \upsilon_1^j} \upsilon_1^j \right\} \\ &+ \sum_{j=1}^{n_u^2} \left\{ \frac{\partial \boldsymbol{J}}{\partial \upsilon_2^j} \dot{\upsilon}_2^j + \frac{\partial \dot{\boldsymbol{J}}}{\partial \upsilon_2^j} \upsilon_2^j \right\}. \end{aligned} \tag{5.5}$$

$\boldsymbol{J}(0,0)$ and $\dot{\boldsymbol{J}}(0,0)$ are the twist Jacobian and its time derivative, according to the actual model.

5.5 Representation uncertainties

The derivatives introduced in the previous Section should be used with care: one is interested in the influence of the uncertainty parameter on the physical motion freedom of the *MO*, i.e., the *vector spaces* spanned by the Jacobians $\boldsymbol{J}, \dot{\boldsymbol{J}}$ and $\boldsymbol{G}$. On the other hand, the partial derivative of a Jacobian matrix with respect to an uncertainty parameter represents

a change in the *representation of a basis* for these vector spaces. It is important to realize that a change in the basis does *not* necessarily imply a change in the vector space spanned by this basis. Hence, blindly interpreting the fact that $\partial \boldsymbol{J}/\partial v$ is not identically zero as an indication that the instantaneous motion freedom modelled by $\boldsymbol{J}$ changes when v changes, can introduce v as a false uncertainty. This mistake occurs whenever a given physical contact situation can have multiple mathematical representations that are all a function of the parameter v. Therefore, such a parameter v is called a *representation uncertainty.* However, it is easy to detect that v is a representation uncertainty: in this case $\partial \boldsymbol{J}/\partial v$ lies completely in span($\boldsymbol{J}$). Indeed, Eq. (5.1) then shows that span($\boldsymbol{J}(v)$) = span($\boldsymbol{J}$).

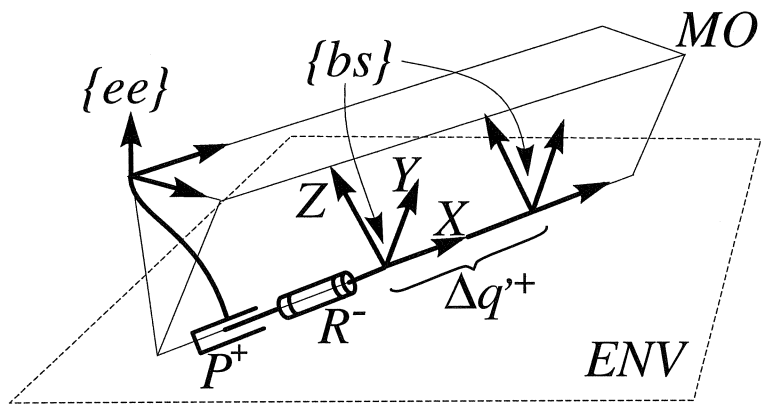


FIGURE 5.5 : *Representation uncertainty for the edge-face contact.*

Some examples from Chapter 4:

Edge-face contact. The uncertainty in the position of the contact frame along the edge of a *edge-face* contact is a representation uncertainty, since the origin of the frame can lie anywhere on the edge without changing the twist system, FIG 5.5. The twist Jacobian has four independent degrees-of-freedom; $\boldsymbol{J}'^{-}$ and $\boldsymbol{J}^{+}$ are dependent since they both represent a translation along the X axis, i.e. the direction of the edge:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}'^{-} & \boldsymbol{J}'^{+} & \boldsymbol{J}^{r} & \boldsymbol{J}^{-} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The joint parameter q'^{+} moves the base frame along the edge, and

$$\frac{\partial \boldsymbol{J}}{\partial q'^{+}} = \boldsymbol{S}_{\Delta}\left(\boldsymbol{J}'^{+}\right)\boldsymbol{J} \overset{\text{Table 3.2}}{=} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

This derivative belongs to span($\boldsymbol{J}$), and hence q'^{+} is a representation uncertainty.

Second order uncertainties. Second order uncertainties are representation uncertainties for the first order twist and wrench systems. Physically, this means that a change in a radius of curvature in the virtual manipulator does not influence the possible instantaneous velocities of the *MO*, neither the ideal contact forces. (However, a change in the curvature parameters does change the nominal accelerations of the *MO*, hence second order uncertainties are not representation uncertainties for $\dot{\boldsymbol{J}}$.) This fact is easily proved: for the general *surface-surface* contact, a second order uncertainty can be modelled by the "joint angle" $v^2 = r$ of a prismatic joint P along the contact normal, FIG. 5.3. To calculate the derivative of the Jacobian with respect to r runs along the lines developed in Chapter 3 and Appendix A, with $\boldsymbol{J}^P$ representing a translational velocity along the contact normal. i.e., $\boldsymbol{J}^P = [0\ 0\ 0\ 0\ 0\ 1]^T$. Hence:

$$
\begin{aligned}
\frac{\partial \boldsymbol{J}}{\partial r} &= \boldsymbol{S}_{\Delta}\left(\boldsymbol{J}^P\right)\boldsymbol{J} \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s & c & 0 & 0 & 1 \\ -c & s & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -r'^{+}c & r'^{-}s & 0 & -r^{+} & 0 \\ -r'^{+}s & -r'^{-}c & 0 & 0 & r^{-} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ c & -s & 0 & -1 & 0 \\ s & c & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
\end{aligned}
$$

c and s are the cosine and sine of the angle between the principal directions of $\{pri\}$ and $\{pri'\}$. This derivative is not identically zero, but its columns belong to span($\boldsymbol{J}$). Hence, r is a representation uncertainty for $\boldsymbol{J}$.

This fact allows to simplify the Taylor series expansions (5.4) and (5.5), because $\partial \boldsymbol{J}/\partial v_2^j = 0$. Hence, after rearrangement of some terms, they reduce to:

$$
\boldsymbol{J} = \boldsymbol{J}(0,0) + \sum_{j=1}^{n_u^1} \frac{\partial \boldsymbol{J}}{\partial v_1^j}\, v_1^j, \tag{5.6}
$$

$$
\begin{aligned}
\dot{\boldsymbol{J}} = {} & \frac{\partial}{\partial t}\left\{\boldsymbol{J}(0,0) + \sum_{j=1}^{n_u^1} \frac{\partial \boldsymbol{J}}{\partial v_1^j}\, v_1^j\right\} \\
& + \sum_{j=1}^{n_u^1} \frac{\partial \boldsymbol{J}}{\partial v_1^j} \dot{v}_j^1 + \sum_{j=1}^{n_u^2} \frac{\partial \dot{\boldsymbol{J}}}{\partial v_2^j} v_2^j.
\end{aligned} \tag{5.7}
$$

Thus, the first order motion freedom of a contact is disturbed by first order uncertainties only. The time rate equation (5.7) falls apart in three contributions:

1. The first order update of Eq. (5.6) changes the nominal Jacobian; hence, first order tracking should be performed before the nominal time derivative of $\boldsymbol{J}$ is calculated.
2. The second part of Eq. (5.6) is due to time changes in the first order uncertainties.
3. The last part of Eq. (5.7) is due to second order uncertainties.

5.6 Multiple contacts

The previous Sections developed the virtual uncertainty manipulator for one single contact only, and described also how its mathematical representation, i.e., its Jacobians, is influenced by the uncertainties. For a contact situation with multiple contacts at the same time, nothing changes in the kinematic model of each individual contact. However, the motion freedom spaces corresponding to the different contacts are *coupled*. Intuitively speaking, more contacts reduce the uncertainty. In fact, this is the main motivation for (active or passive) compliant motion execution of assembly tasks in the first place: use contacts to guide the *MO* towards its (inaccurately known) goal position. So, also speaking intuitively, it should be possible to use fewer uncertainty parameters in a motion constraint with more contacts. However, this statement is only true for parallel uncertainty manipulators between the *same* object reference frames. This Section presents a systematic procedure to reduce the number of first order uncertainty joints in the model.

First order uncertainties correspond to inaccurate knowledge of the position of the origin of the virtual contact manipulator's principal frames with respect to the robot's end effector frame. Section 5.3 used the joint angles q^+, q^-, q'^+ and q'^- to model these uncertainties in the case of a general *surface-surface* contact. Other contact cases can use different types of kinematic uncertainty models. The "easiest" way could even be to introduce a six degrees-of-freedom virtual uncertainty manipulator between the two frames for every contact. In this way, one quickly ends up with a large set of uncertainty parameters. This is not exactly an advantage for the subsequent identification of the uncertainties. Hence, a systematic procedure to eliminate redundant joints from the virtual uncertainty manipulators would be welcome. Such a procedure looks for dependencies between the uncertainty motion freedom spaces spanned by the different virtual uncertainty manipulators, and deletes the dependent joints from the model. The reduction procedure presented below is based on a straightforward mobility analysis. It is illustrated by the *peg-on-hole* contact situation, FIGS. 4.6 and 4.7. Assume that the position of the peg with respect to the robot end effector is exactly known, i.e., the virtual manipulator's "end effector" frame $\{ee\}$ is known with respect to the robot. However, the positions of each contact with respect to $\{ee\}$ are not, and a six-degrees-of-freedom uncertainty manipulator is introduced between $\{ee\}$ and $\{bs\}$, the "base" reference frame at the centre of the hole. Then, the following five steps constitute the uncertainty joint reduction procedure:

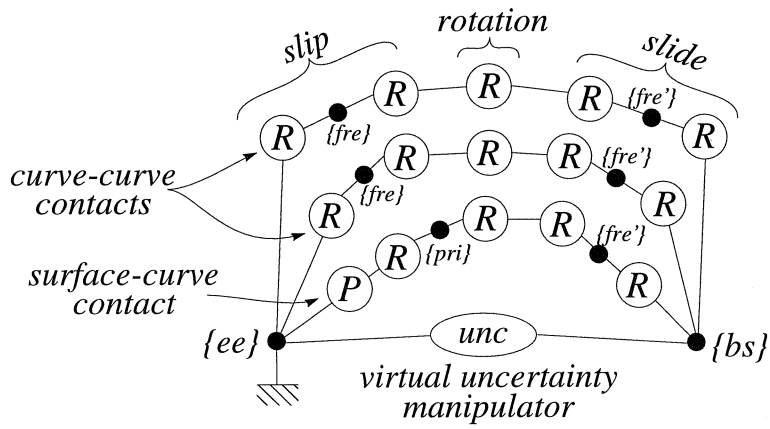


FIGURE 5.6 : *Peg-on-hole: topology of total kinematic chain.* R indicates a revolute joint, P a prismatic joint; the bullets are the geometric reference frames on the virtual contact manipulators.

1. Construct the total kinematic chain consisting of: 1) the robot; 2) the nominal virtual contact manipulator(s); and 3) the uncertainty joints.
   FIGURE 4.7 depicts the nominal kinematic models for the peg-on-hole contact. Since the curves involved in the contacts are prefectly modelled by these nominal kinematic models, the uncertainties correspond to uncertainties in the joint positions of these models.

2. Measure the robot joint angles. This eliminates the robot's degrees-of-freedom from the total kinematic chain, because they are now fixed to known values.
   FIGURE 5.6 is the result for the peg-on-hole contact situation: $\{ee\}$ is known with respect to the absolute world; $\{bs\}$ is not.

3. Find all kinematic chains between $\{ee\}$ and $\{bs\}$, consisting of both nominal and uncertainty joints.
   In the peg-on-hole case, there are four parallel kinematic chains: one for each contact, and one uncertainty chain.

4. The uncertainty motion freedom vector space between the two frames is the *intersection* of the motion freedom vector spaces of all these chains.
   For the peg-on-hole case, each of the contacts reduces one degree of freedom from the six-dimensional uncertainty space. Hence, three uncertainty degrees of freedom remain.

5. Select a minimum set of uncertainty joints which span this intersection.
   For the peg-on-hole situation, three uncertainty joints remain, from the initial six (or even eighteen, in the most simplistic case of introducing a separate six degrees-of-freedom uncertainty manipulator for each contact!). With some insight, it is clear that they can be assigned to the three following joints already present in the virtual

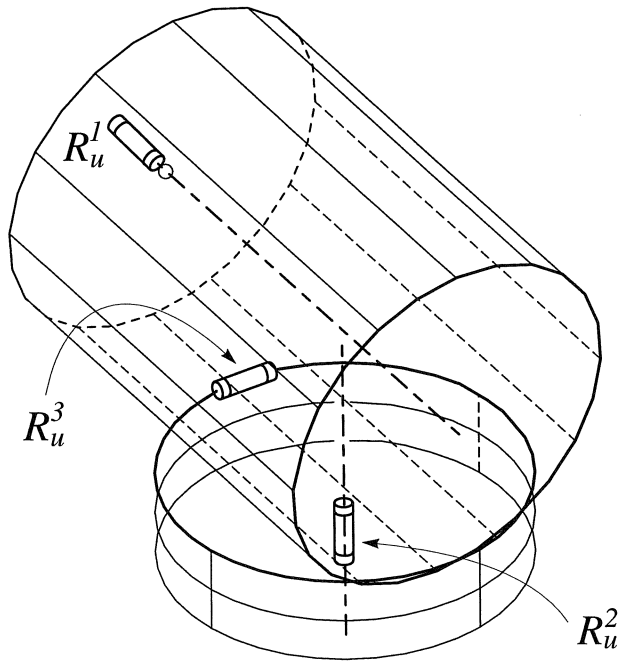


FIGURE 5.7 : *Peg-on-hole: minimum uncertainty joints*: R_u^1, R_u^2 and R_u^3.

contact manipulators, FIG. 5.7: 1) R_u^1, which is a rotation about the peg's axis; 2) R_u^2, which is a rotation about the hole's axis; and 3) R_u^3, i.e., the *alignment error* between peg and hole, which is a rotation about the revolute joint tangent to the hole's rim at the *surface-curve* contact. In this contact situation, these three joints also represent the nominal motion freedom of the peg with respect to the hole, Section 4.4.3.

Such an a priori reduction of the uncertainties —if it is possible— results in a high efficiency gain during on-line identification since it reduces the dimension of the uncertainty search space.

5.7 Conclusions

This Chapter introduced geometric uncertainty parameters into the nominal kinematic contact models of Chapter 4. These uncertainties are classified in two complementary groups: time invariant vs. time varying on the one hand, and first order vs. second order on the other hand.

First order uncertainties represent inaccurate knowledge about the position of the contact point and the direction of the contact normal; they influence the instantaneous motion freedom space spanned by the twist and wrench Jacobians $\boldsymbol{J}$ and $\boldsymbol{G}$. Second order uncertainties represent inaccurate knowledge about the curvature of the contacting surfaces: directions of minimum and maximum curvature, and the corresponding curvature radii. They do not influence the span of $\boldsymbol{J}$ and $\boldsymbol{G}$, but they do change the second order motion

freedom models.

Time invariant and time varying uncertainties are applied in different contexts. Time invariant uncertainties represent a priori unknown but constant geometric parameters, such as the relative position and orientation of two reference frames in the system which are fixed with respect to each other; they are modelled by virtual uncertainty manipulators that do not exist yet in the nominal virtual contact manipulators. These uncertainties have to be identified only once, and this "calibration" can (and most often should) be done before the compliant motion task. Time varying uncertainties represent the continuously changing local contact geometry, due to the relative motion of smooth surfaces or curves over each other; they are modelled as small errors in the nominal virtual contact manipulators. The second order uncertainties require the introduction of prismatic joints along the radii of curvature to account for changes in a radius of curvature.

In principle, the contact type, as listed in TABLE 4.2, suffices to determine the generic set of uncertainty parameters. Any extra a priori knowledge about the contact geometry reduces the set of uncertainty joints, and hence simplifies the on-line identification.

The mathematical description of the influence of the uncertainties on the nominal motion freedom of the MO is given by the derivatives of the Jacobian matrices $\boldsymbol{J}, \boldsymbol{G}$ and $\dot{\boldsymbol{J}}$ with respect to the uncertainty joint angles. Straightforward application of the derivative equations developed in Chapter 3 can lead to false uncertainties, called "representation uncertainties." A simple check controls whether an uncertainty belongs to this class or not.

When multiple contacts occur at the same time, nothing changes from the uncertainty modelling point of view. However, if two or more contacts link the same reference frames on the objects, the number of uncertainty joints can be reduced. This systematic reduction procedure has been applied to the example of the "peg-on-hole" contact situation.

Chapter 6

Identification

6.1 Introduction

Identification of system parameters is a very rich and extended research area. However, there is no such thing as *the* general identification approach. Therefore, this Chapter introduces the principles behind the approach adopted in this text to identify the geometric uncertainties modelled in the previous Chapter. That Chapter defined the uncertainty "motion freedom" space as the vector space spanned by the partial derivatives of the Jacobian matrices with respect to the joint angles in the virtual uncertainty manipulators.

First of all, Section 6.2 introduces the concepts of *reciprocity* and *consistency* as the physical constraints to which the model, the measurements and the uncertainty parameters must obey in order to correspond to a real *compliant* motion.

Sections 6.3 and 6.4 apply these concepts to derive identification equations for first and second order uncertainties. The inputs to the identification equations are either the measured velocities and accelerations, or the measured forces. Hence, these identification approaches are called *twist based* and *wrench based*, respectively. Reciprocity as well as consistency are used as fundamental identification principles. Both approaches are proven to be equivalent in Section 6.5.

Section 6.6 compares the reciprocity and consistency based approaches from a practical point of view. It also discusses when the identification equations derived in the previous Sections can be applied, how they should be interpreted, what their limits are and by what phenomena they are disturbed. This Section also introduces the concept of *monitoring* equations. These complement the identification equations, in that they can be used to detect when the model of the current contact is no longer valid.

Section 6.7 illustrates the theoretical twist based and wrench based approaches from Sections 6.3 and 6.4 by means of the 2D contour following task. Both first and second order identification equations are deduced and intuitively interpreted when possible. Section 6.8 lists the first order identification equations for the most common polyhedral contact situations.

6.2 Reciprocity and consistency

A motion of the *MO* is called *(first order) reciprocal*, Ohwovoriole and Roth (1981), if it maintains the contact up to first order. This means that the contact point moves instantaneously in the tangent plane. The reciprocity constraint links the *MO*'s motion — i.e., its *twist* $\mathsf{t} = [\boldsymbol{\omega}^T \ \boldsymbol{v}^T]^T$— to the ideal reaction force —i.e., the *wrench* $\mathsf{w} = [\boldsymbol{f}^T \ \boldsymbol{m}^T]^T$— generated by the contact, by requiring that t produces no *power* against w:

$$\boldsymbol{m}^T\boldsymbol{\omega} + \boldsymbol{f}^T\boldsymbol{v} = 0, \text{ or } \mathsf{t}^T\tilde{\boldsymbol{\Delta}}\mathsf{w} = 0, \tag{6.1}$$

$$\text{with } \tilde{\boldsymbol{\Delta}} = \begin{bmatrix} \boldsymbol{O}_{3\times3} & \boldsymbol{I}_{3\times3} \\ \boldsymbol{I}_{3\times3} & \boldsymbol{O}_{3\times3} \end{bmatrix}. \tag{6.2}$$

After elimination of the time dimension, this reciprocity relation represents the *(virtual) work* between the contact wrench and an infinitesimal displacement twist t_Δ of the *MO*. The relation remains valid for multiple contacts: in this case t is a motion of the *MO* that makes all contact points move instantaneously in their local tangent planes, and w is the resultant of the reaction forces in each contact.

Reciprocity straightforwardly extends to accelerations and time rates of the contact forces, by taking the time derivative of Eq. (6.1):

$$\frac{\partial}{\partial t}\left(\boldsymbol{m}^T\boldsymbol{\omega} + \boldsymbol{f}^T\boldsymbol{v}\right) = 0, \text{ or } \frac{\partial}{\partial t}\left(\mathsf{t}^T\tilde{\boldsymbol{\Delta}}\mathsf{w}\right) = 0. \tag{6.3}$$

A motion is reciprocal to the constraints up to *second order* if it fulfills both requirements (6.1) and (6.3).

Consistency requires that the motion and reaction force models are compatible with the motion and the reaction force measurements, respectively, Bruyninckx, De Schutter & Dutré (1993). In terms of the kinematic models this means that: 1) the velocities and accelerations of all joints in the virtual contact manipulators are consistent with the executed Cartesian motion of the *MO* (these are the *velocity* and *acceleration closure* requirements); 2) the reaction forces at the joints of the virtual contact manipulators, caused by the (ideal) contact forces, are completely absorbed by the "mechanical" structure of the manipulators, and hence no driving torques are needed at the joints to resist the contact force.

First order consistency links the measured twist t_m and wrench w_m to the twist and wrench Jacobians by linear equations:

$$\forall i = 1, \ldots n_c, \exists \dot{\boldsymbol{q}}^i = \left[\dot{q}^i_1 \ldots \dot{q}^i_{n^f_i}\right]^T \ : \mathsf{t}_m = \boldsymbol{J}^i\dot{\boldsymbol{q}}^i, \tag{6.4}$$

$$\exists \boldsymbol{\mathcal{T}} = \left[\boldsymbol{\mathcal{T}}^1 \ldots \boldsymbol{\mathcal{T}}^{n_c}\right]^T, \boldsymbol{\mathcal{T}}^i = [\tau^i_1 \ldots \tau^i_{6-n^f_i}]^T \ : \mathsf{w}_m = \boldsymbol{G}\boldsymbol{\mathcal{T}}. \tag{6.5}$$

n_c is the number of parallel constraints. n^f_i is the number of degrees of motion freedom in constraint i; n^f_i can be smaller than five, depending on the contact type, TABLE 4.2.

$\boldsymbol{J}^i$ is the twist Jacobian of the ith constraint. $\dot{q}^i_j$ is the velocity of the jth joint in the ith virtual contact manipulator. $\boldsymbol{G}$ is the wrench Jacobian of the total constraint, i.e., it is the *concatenation* of all $\boldsymbol{G}^i$: $\boldsymbol{G} = \left[\boldsymbol{G}^1 \ \ldots \ \boldsymbol{G}^{n_c}\right]$. $\mathcal{T}^i = [\tau^i_1 \ldots \tau^i_{6-n^f_i}]^T$ is the vector of contact force and/or torque components in constraint i.

Hyperstaticity. Equation (6.5) becomes singular for a *hyperstatic* constraint. Hyperstaticity represents the fact that not all contact forces and torques in the total constraint are independent. Practically, it means that the force or torque components of an individual contact cannot be derived unambiguously from the measured wrench w_m. This situation occurs often in the case of multiple contacts. Moreover, the presence of friction makes it even more difficult to estimate the individual ideal reaction forces.

6.3 Consistency based identification

This Section derives identification equations from the concept of consistency. n_c contacts in parallel are considered, as well as first and second order uncertainties.

6.3.1 First order identification

Filling in the first order expansions of the Jacobians, Eq. (5.6), in the consistency conditions, Eqs. (6.4)-(6.5), give *non-linear* equations in the unknowns v^i_j, $\dot{q}^i_k$, and τ^l, and with the *measured* twist t_m and wrench w_m as known inputs:

$$\mathsf{t}_m = \left\{ \boldsymbol{J}^i(\mathcal{V}^i = 0) + \sum_{j=1}^{n^i_u} \frac{\partial \boldsymbol{J}^i}{\partial v^i_j} v^i_j \right\} \dot{\boldsymbol{q}}^i, \qquad \forall i = 1, \ldots, n_c, \tag{6.6}$$

$$\mathsf{w}_m = \left\{ \boldsymbol{G}(\mathcal{V} = 0) + \sum_{v_j \in \mathcal{V}} \frac{\partial \boldsymbol{G}}{\partial v_j} v_j \right\} \mathcal{T}. \tag{6.7}$$

$\mathcal{V}^i = \{v^i_1, \ldots, v^i_{n^i_u}\}$ is the set of first order uncertainty parameters for constraint i; $\mathcal{V}$ is the first order uncertainty parameter set for all constraints: $\mathcal{V} = \cup_{i=1}^{n_c} \mathcal{V}^i$.

Equation (6.6) represents six *twist based identification* equations for constraint i. Their physical interpretation is that one tries to make the kinematic model consistent with the measured twist t_m of the *MO* by changing 1) the basis of the motion freedom space, $\boldsymbol{J}^i$, and/or 2) the velocities of the joints in the virtual contact manipulator. Twist based identification does not introduce *coupling* between the different contact models, because making the kinematic model in one contact consistent with the measured twist t_m of the *MO* does not change this measured twist as seen by the other contacts.

Equation (6.7) represents six *wrench based* identification equations for the total constraint. Their physical interpretation is that one tries to make the kinematic model consistent with the measured wrench w_m exerted on the *MO* by changing 1) the basis of the space of ideal reaction forces, and/or 2) the magnitudes of the reaction forces at all contacts.

Wrench based identification does introduce coupling between the different parallel contact: changing the kinematic model in one contact —i.e., the direction or the magnitude of the ideal contact force at this contact— affects the part of the measured wrench that has to be accounted for by the other contacts.

Not only the parameters v_j^i are of interest; also the parameters $\dot{q}_k^i$ and τ^l give valuable identification information: $\dot{q}_k^i$ indicates how fast a particular joint in the virtual manipulator moves (this directly affects the evolution of the nominal model!); τ^l gives the reaction force at one specific contact. The summation in Eq. (6.7) runs over all uncertainty parameters in the total constraint model; this number need not be equal to $\sum_{i=1}^{n_c} n_u^i$, if reduction of uncertainty parameters is possible, Section 5.6.

6.3.2 Second order identification

Second order identification equations are derived from the discrepancy between, on the one hand, the nominal time evolution of the velocity closure relation, and, on the other hand, the measured acceleration $\dot{\mathsf{t}}_m$ of the *MO*. For example, twist based identification results in the following set of non-linear equations:

$$\dot{\mathsf{t}}_m = \frac{\partial \left(\boldsymbol{J}^i(\mathcal{V}^i)\ \dot{\boldsymbol{q}}^i \right)}{\partial t} = \dot{\boldsymbol{J}}^i(\mathcal{V}^i)\ \dot{\boldsymbol{q}}^i + \boldsymbol{J}^i(\mathcal{V}^i)\ \ddot{\boldsymbol{q}}^i. \tag{6.8}$$

The physical interpretation is very similar to the first order case: the measured acceleration of the *MO* is made consistent with the model by changing 1) the bases of the first and second order motion freedom spaces, $\boldsymbol{J}^i$ and $\dot{\boldsymbol{J}}^i$, and/or 2) the velocities and accelerations of the joints in the virtual contact manipulators, $\dot{q}_j^i$ and $\ddot{q}_j^i$.

The dependencies of the Jacobians on the uncertainties are described by Eqs. (5.6) and (5.7). The first term in this last equation indicates that second order identification is only meaningful if first order identification is performed first.

6.4 Reciprocity based identification

This Section derives identification equations from the concept of reciprocity. n_c contacts in parallel are considered, as well as first and second order uncertainties.

6.4.1 First order identification

First order reciprocity means that the measured twist t_m and wrench w_m are linked to the Jacobians in such a way that, Bruyninckx & De Schutter (1992):

1. The power generated by the measured twist against every basis wrench in the wrench Jacobian $\boldsymbol{G}^i$ of the ith constraint vanishes:

$$\mathsf{t}_m^T\ \widetilde{\boldsymbol{\Delta}}\ \boldsymbol{G}^i(\mathcal{V}^i) = 0,\ \forall i = 1, \ldots, n_c. \tag{6.9}$$

2. The power generated by the measured wrench against every basis twist in the twist Jacobian $\boldsymbol{J}$ of the total constraint vanishes:

$$\mathsf{w}_m^T \, \tilde{\boldsymbol{\Delta}} \, \boldsymbol{J}(\upsilon) = 0. \tag{6.10}$$

Remark the duality with the consistency based approach: $\boldsymbol{G}^i$ is the wrench Jacobian of the ith contact; $\boldsymbol{J}$ is the twist Jacobian of the total constraint, i.e., it is a basis for the *intersection* of the twist systems of all constraints: $\text{span}(\boldsymbol{J}) = \cap_i \left(\text{span}\left(\boldsymbol{J}^i\right)\right)$. Reciprocity conditions, similar to Eqs. (6.9) and (6.10) exist between the Jacobians of *any* motion constraint:

$$\left(\boldsymbol{J}^i\right)^T \tilde{\boldsymbol{\Delta}} \boldsymbol{G}^i = 0, \ \forall i = 1, \ldots, n_c. \tag{6.11}$$

This condition is valid for all values of the uncertainty parameters: $\left(\boldsymbol{J}^i(\upsilon^i)\right)^T \tilde{\boldsymbol{\Delta}} \boldsymbol{G}^i(\upsilon^i) = 0, \forall i = 1, \ldots, n_c$. Filling in the Jacobians' series expansions, Eq. (5.6), in the reciprocity conditions, Eqs. (6.9)-(6.10), gives *linear* identification equations:

$$\mathsf{t}_m^T \, \tilde{\boldsymbol{\Delta}} \, \boldsymbol{G}^i(\upsilon^i = 0) \ + \ \sum_{j=1}^{n_u^i} \mathsf{t}_m^T \, \tilde{\boldsymbol{\Delta}} \, \frac{\partial \boldsymbol{G}^i}{\partial \upsilon_j^i} \upsilon_j^i \ = \ 0, \tag{6.12}$$

$$\mathsf{w}_m^T \, \tilde{\boldsymbol{\Delta}} \, \boldsymbol{J}(\upsilon = 0) \ + \ \sum_{\upsilon_j \in \upsilon} \mathsf{w}_m^T \, \tilde{\boldsymbol{\Delta}} \, \frac{\partial \boldsymbol{J}}{\partial \upsilon_j} \upsilon_j \ = \ 0. \tag{6.13}$$

These equations' physical interpretation is that the motion constraint model is adapted, by means of the uncertainty parameters υ, until the above-mentioned reciprocal power between measured and modelled twists and wrenches vanishes.

Equation (6.12) represents $6 - n_f^i$ twist based identification equations for constraint i; Eq. (6.13) is the wrench based counterpart for the total constraint, giving $\dim(\boldsymbol{J})$ equations. The summation in Eq. (6.13) runs over all uncertainty parameters in the total constraint model; again, this number need not be equal to $\sum_{i=1}^{n_c} n_u^i$.

6.4.2 Second order identification

Linear second order identification equations result from the time rate of the reciprocity conditions. For example, twist based identification gives:

$$0 = \frac{\partial \left(\mathsf{t}_m^T \, \tilde{\boldsymbol{\Delta}} \, \boldsymbol{G}^i(\upsilon)\right)}{\partial t} = \dot{\mathsf{t}}_m^T \, \tilde{\boldsymbol{\Delta}} \, \boldsymbol{G}^i(\upsilon) + \mathsf{t}_m^T \, \tilde{\boldsymbol{\Delta}} \, \dot{\boldsymbol{G}}^i(\upsilon). \tag{6.14}$$

Similar remarks as for consistency based second order identification apply, except that the twist and wrench system parameters $\dot{q}^i$ and $\mathcal{T}^i$ do not appear in the reciprocity based identification equations.

6.5 Equivalence between Reciprocity and Consistency

This Section describes how the first order consistency twist based identification equations (6.4) can be transformed into the first order reciprocity twist based identification equations (6.9). A similar transformation applies to the first order wrench based identification equations.

Multiplying the transpose of the first order twist based consistency equation (6.6) by $\tilde{\boldsymbol{\Delta}}\boldsymbol{G}^i(\upsilon^i = 0)$ and applying the reciprocity condition (6.11) results in:

$$\mathsf{t}_m^T \tilde{\boldsymbol{\Delta}} \boldsymbol{G}^i(\upsilon^i = 0) = \sum_{j=1}^{n_u^i} \upsilon_j^i \left(\dot{\boldsymbol{q}}^i\right)^T \left(\frac{\partial \boldsymbol{J}^i}{\partial \upsilon_j^i}\right)^T \tilde{\boldsymbol{\Delta}} \boldsymbol{G}^i(\upsilon^i = 0). \tag{6.15}$$

Expanding the reciprocity condition, Eq. (6.11), in the uncertainty parameters υ_j^i gives:

$$\begin{aligned} 0 &= \left(\boldsymbol{J}(\upsilon^i)\right)^T \tilde{\boldsymbol{\Delta}}\, \boldsymbol{G}^i(\upsilon^i) \\ &= \left(\boldsymbol{J}^i(\upsilon^i = 0) + \sum_{j=1}^{n_u^i} \frac{\partial \boldsymbol{J}^i}{\partial \upsilon_j^i} \upsilon_j^i\right)^T \tilde{\boldsymbol{\Delta}} \left(\boldsymbol{G}^i(\upsilon^i = 0) + \sum_{j=1}^{n_u^i} \frac{\partial \boldsymbol{G}^i}{\partial \upsilon_j^i} \upsilon_j^i\right). \end{aligned}$$

Since $\left(\boldsymbol{J}^i(\upsilon^i = 0)\right)^T \tilde{\boldsymbol{\Delta}}\, \boldsymbol{G}^i(\upsilon^i = 0) = 0$, this yields:

$$\sum_{j=1}^{n_u^i} \upsilon_j^i \left(\frac{\partial \boldsymbol{J}^i}{\partial \upsilon_j^i}\right)^T \tilde{\boldsymbol{\Delta}} \boldsymbol{G}^i(\upsilon^i = 0) = -\left(\boldsymbol{J}^i(\upsilon^i)\right)^T \tilde{\boldsymbol{\Delta}} \sum_{j=1}^{n_u^i} \frac{\partial \boldsymbol{G}^i}{\partial \upsilon_j^i} \upsilon_j^i. \tag{6.16}$$

Using Eq. (6.16) in Eq. (6.15) gives:

$$\mathsf{t}_m^T \tilde{\boldsymbol{\Delta}} \boldsymbol{G}^i(\upsilon^i = 0) = -\sum_{j=1}^{n_u^i} \upsilon_j^i \left(\dot{\boldsymbol{q}}^i\right)^T \left(\boldsymbol{J}^i(\upsilon^i)\right)^T \tilde{\boldsymbol{\Delta}} \frac{\partial \boldsymbol{G}^i}{\partial \upsilon_j^i}. \tag{6.17}$$

This is equivalent to the first order twist based reciprocity equation (6.12), with $\boldsymbol{J}^i\dot{\boldsymbol{q}}^i$ replaced by t_m, Eq. (6.4). A similar operation transforms the first order wrench based consistency equations (6.7) into the first order wrench based reciprocity equations (6.13).

6.6 Identification in practice

This Section summarizes the results of the reciprocity and consistency based identification approaches, as developed in the previous Sections, and discusses their practical usefulness.

Reciprocity	**Consistency**
Identification equations are *linear* in the uncertainty parameters.	Identification equations are *non linear* in the uncertainty parameters; i.e., also the virtual joint velocities or the individual contact forces are unknowns.
$\dim(\boldsymbol{J})$ *wrench based* equations for *total* constraint.	6 *wrench based* equations for *total* constraint.
$6-n_f^i$ *twist based* equations for constraint i.	6 *twist based* equations for constraint i.
Identification and tracking: 1. Calculate uncertainties from the linear equations (using a *one-step* algorithm). 2. Adapt model, i.e., it is now consistent with the measurements. 3. Solve for virtual joint velocities and individual contact forces.	Identification and tracking: 1. Solve for uncertainties, virtual joint velocities and individual contact forces from non linear equations. In general, this involves an *iterative* approach. However, in the intermediate iterations, the joint velocities and the individual contact forces must be solved from inconsistent equations. 2. Adapt model.

The comparison above shows that identification and tracking are coupled in the consistency based approach. However, this coupling is easily eliminated by the procedure of the previous Section. Hence, in practice, both approaches differ only in the initial definition of the system of equations to be solved, but lead to identical *identification* results.

6.6.1 Monitoring equations

The series expansions of the Jacobians, Eqs. (5.6) and (5.7), are fundamental to both the reciprocity and the consistency based identification approaches. Since only the first order terms in these Taylor series are used, it is implicitly assumed that the geometric uncertainties remain small. This is an acceptable assumption for smoothly varying motion constraints:

1. When used on-line, the identification equations continuously produce estimates of how the model should be corrected. These estimates serve as inputs to a tracking module, which makes sure that the discrepancies in the model never grow large. Unless, of course, the initial model is already inaccurate, or a "catastrophic" change in the contact situation occurs.

2. Large time varying uncertainties imply that the model is very wrong about where and how the contacts take place. This situation is handled better by a set of time invariant uncertainties and a preceding *calibration* within a known environment (if the uncertainty lies with the manipulated object), or with a known manipulated object (if the environment is uncertain). The task would probably fail anyway because the motion specification generated by the model is totally inappropriate.

If the motion constraint varies smoothly, the outcome of the identification will vary smoothly too. Hence, a transition to another contact type most often produces a noticeable jump in the identification results. By continuously keeping an eye on the identification results, the current assumption about the *type* of the contact can be *monitored*. This means that the fact that some of the identification results grow "too large" is interpreted as a change in the current contact configuration. ("Large" and "small" are very relative indications. Indeed, the robustness against uncertainties of the total compliant motion system depends on much more than only the solvability and numerical conditioning of the identification equations.)

Besides the identification results, other monitoring information is available too. At first sight, consistency based identification seems to lead to overly complicated equations. In addition, it always gives *six* equations. But some of these six equations turn out to be identities. In fact, these identities are only exactly satisfied *after* the model is adapted and made consistent. Otherwise, non-zero residuals remain. As for the identification results, these residuals should be small; if one or more of them suddenly grow large, the type of the contact has probably changed. Moreover, these changes in the consistency residuals can give some partial information about the new contact type. The *vertex-surface* contact, FIG. 6.1, yields a simple illustration of this idea. (More examples follow in Section 6.8.) If the modelled orientation of the Z axis is not far from the real contact normal direction —i.e., the uncertainties Δq^- and Δq^+ are small— only small force components along X and/or Y should be measured, together with a significant Z component. Hence, increasing X or Y force components indicate a transition to a new contact situation in which the translation of the MO in the corresponding direction is also constrained. Similarly, if the Z force suddenly disappears, then the MO has probably reached the edge of the constraint surface.

6.6.2 Identifiability

The following paragraphs discuss two important factors that determine whether the modelled uncertainties are identifiable, i.e., whether the identification equations for a given contact are solvable: 1) the *mathematical properties* of the set of equations, and 2) the *disturbances* that can be expected from the real world.

Mathematical properties. The mathematical solvability of the set of identification equations has many aspects:

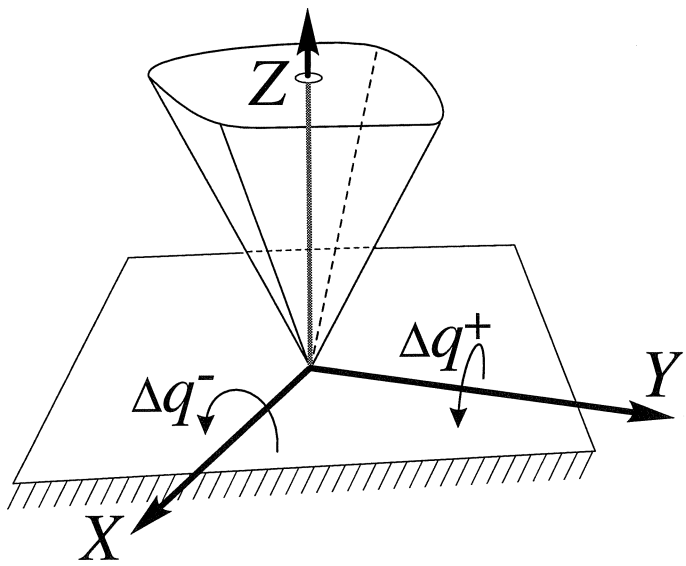


FIGURE 6.1 : Vertex-surface contact. The difference between the modelled orientation of the Z axis and the real contact normal is modelled by the uncertainties Δq^- and Δq^+.

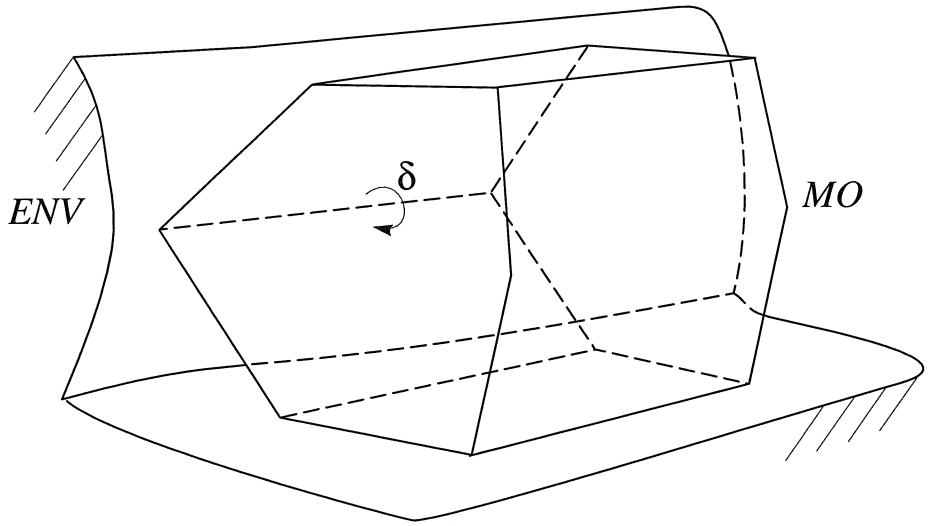


FIGURE 6.2 : *Example of unidentifiable uncertainty due to multiple contacts.* The contact normal at the left-hand side edge of the *MO* cannot be identified, since the *face-face* contact at the bottom of the *MO* prevents the "excitation" of the uncertainty δ.

1. The coefficient of an uncertainty v in the set of identification equations can be *identically zero*. v is then *unidentifiable*. This is, by definition, the case for a representation uncertainty. An example of an unidentifiable real uncertainty is shown in FIG.6.2.

2. The coefficient of v can instantaneously be made zero by a coincidental relation between the components of the twist or wrench measurements. v is then called *conditionally unidentifiable*, since it is only momentarily insufficiently "excited" by the current motion or contact force.

3. The set of equations can be *underdetermined*, i.e., there are more independent unknowns (i.e., uncertainties) than equations. This means that not enough identification information is available through the measurements; i.e., the uncertainty model is *insufficiently excited*. For example, when following a smooth surface with a vertex-like MO, twist based identification yields only one single identification equation: it can instantaneously give information about the tangent in the direction of motion, but the tangent in the transverse direction is conditionally unidentifiable. It could be identified by moving in that direction, but then the other direction has become unidentifiable.

4. The set of equations can be *overdetermined*, i.e., there are more independent equations than unknowns. If this set is *inconsistent*, then too few uncertainty parameters have been introduced. If it is consistent, the equations are not independent and hence the uncertainty parameters can be found unambiguously.

5. Second order identification can only be done after first order identification has been performed successfully. Both identification levels could also be scheduled at different frequencies.

6. The identification equations of this Chapter have been derived in the first place for *instantaneous* and *recursive* application to *time varying* motion constraints. Time invariant uncertainties are better identified by a *batch identification* approach, especially if the set of equations is underdetermined. This is indeed the classical approach for general system identification: many uncertain time invariant parameters, many measurements in different configurations of the system, followed by some sort of "least squares" identification. However, this approach is not generally applicable to compliant motion tasks, because: 1) few contact types are time invariant; and 2) no natural norm exists for 3D motion or forces (Chapter 2), and hence a least squares approach could lead to non-invariant results.

   Chapter 7 presents a batch identification experiment: the estimation of the radius of an unknown cylinder. Since it is known that the object is cylindrical, the requirement of time invariance is satisfied.

7. If multiple contacts occur simultaneously, the motion freedom at each contact shrinks,

while at the same time the vector space of possible reaction forces grows:

$$\text{span}\,(\boldsymbol{J}) = \bigcap_i \left(\text{span}\left(\boldsymbol{J}^i\right)\right), \quad \text{span}\,(\boldsymbol{G}) = \bigoplus_i \left(\text{span}\left(\boldsymbol{G}^i\right)\right). \tag{6.18}$$

This reduced instantaneous motion freedom increases the chance of obtaining conditionally unidentifiable uncertainties. This is indeed the case for twist based identification if:

$$\frac{\partial \boldsymbol{J}^i}{\partial \upsilon_j} \notin \bigcap_{k \neq i} \left(\text{span}\left(\boldsymbol{J}^k\right)\right), \tag{6.19}$$

i.e., the change in motion freedom at the ith contact, generated by the uncertainty υ_j, is blocked by the other contacts. Hence, there is no way to "excite" the uncertainty υ_j or to observe its influence. Similarly for wrench based identification if:

$$\frac{\partial \boldsymbol{G}^i}{\partial \upsilon_j} \in \bigoplus_{k \neq i} \left(\text{span}\left(\boldsymbol{G}^k\right)\right), \tag{6.20}$$

i.e., the change in reaction force at the ith contact, generated by the uncertainty υ_j, falls within the vector space of reaction forces already spanned by the other contacts. FIGURE 6.2 illustrates this phenomenon.

Disturbances. The following non-ideal effects generate disturbances for the identification process:

1. *Friction* disturbs the force measurements, and hence also the wrench based identification equations. Indeed, these identification equations assume that the *ideal* contact force is measured, but what is really measured is this ideal contact force plus the friction force.

2. *Compliance* of the mechanical structure (robot, fixtures, manipulated object, force sensor, ...) disturbs the twist measurement, and hence the twist based identification equations. Indeed, these identification equations assume that the motion of the *MO* is measured, but what is really measured is this motion plus the deformation of all compliant elements (including robot joints or links) between *MO* and robot base. In addition, compliance might disturb the force measurements since it can induce low-frequency vibrations in the unconstrained motion subspace.

3. *Sensor noise* disturbs the force and velocity measurements, and even more their time derivatives. Moreover, even if the ideal system gives rise to a consistent mathematical model, sensor noise will make it inconsistent.

4. Industrial robot arms and sensors are not very accurate measurement devices. Hence, identification of very complex contact situations, with closely separated contacts — e.g., complex plastic moulds, light switches, watches, Integrated Circuits, ...— is practically infeasible.

5. The combined dynamics of identification, tracking and force control have not been investigated. Hence, no convergence or stability proofs exist.

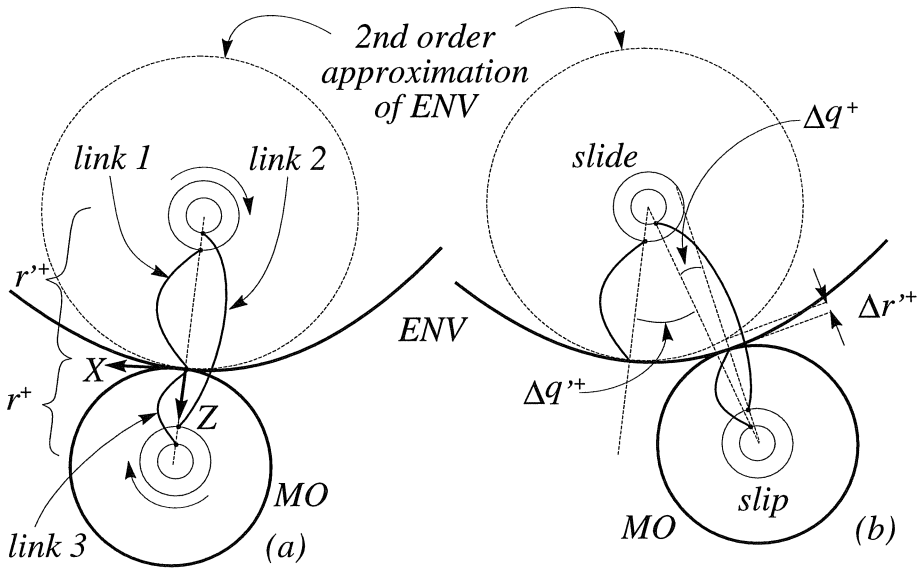


FIGURE 6.3 : *Uncertainties in 2D contour following.* The model assumes contact situation *(a)*, while, in reality, situation *(b)* occurs.

6.7 Example: 2D contour following

FIGURE 1.12 shows 2D contour following with smoothly curved surfaces for both *MO* and *ENV*. The kinematic model approximates this contact situation by two circles with radii r^+ and r'^+ respectively, FIG. 6.3. (The *vertex-surface* contact is a limit case, with $r^+ \rightarrow 0$.) The kinematic model consists of three links and two joints: the first link runs from the principal frame $\{pri\}$ on the *MO* to the slide joint R'^+, i.e., the "base" frame $\{bs\}$ of the virtual contact manipulator is chosen to coincide with $\{pri\}$; the second link directly connects the slide joint R'^+ to the slip joint R^+, since there is no *rotation* joint in the 2D case; and the third link runs from this slip joint R^+ to the "end effector" frame $\{ee\}$, which is also chosen to coincide with $\{pri\}$.

This Section illustrates twist based and wrench based identification, applying *reciprocity* as physical search principle. (The consistency based approach gives exactly the same results, as proved in Section 6.5.) The Jacobians, expressed with respect to the principal frame, Section 4.2, and with the positive directions of the R^+ and R'^+ joints as indicated in FIG. 6.3, are:

$$\boldsymbol{J} = \left[\boldsymbol{J}'^+ \; \boldsymbol{J}^+\right] = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ r'^+ & -r^+ \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6.21}$$

These expressions are valid for all three representations (hybrid, inertial and body-fixed)

introduced in Chapter 3, *if* the base and end effector frames of the virtual contact manipulators instantaneously coincide with the principal frame of the *MO*. In the kinematic model, the first order uncertainties correspond to: 1) Δq^+, the angle error of the slip joint R^+; and 2) $\Delta q'^+$, the angle error of the slide joint R'^+, FIG. 6.3. The uncertainties for second order identification are Δr^+ and $\Delta r'^+$.

The identification equations derived in the previous Sections link *measured* twists and wrenches to the *modelled* motion freedom and constraint spaces. The reciprocity and consistency based identification approaches are valid regardless of how the twists and wrenches are measured, and of what representation is used. This means that the same uncertainty parameters are found. Nevertheless, the identification equations themselves do depend on the representation (hybrid, inertial or body-fixed), and on the chosen measurement reference frames. From TABLE 3.4 it is clear that the hybrid representation results in general in the most "dense" identification equations, i.e., it gives more non-zero coefficients for the uncertainty parameters than the body-fixed or inertial representations. Hence, there is less chance that some uncertainties are unidentifiable because they appear with zero coefficient in the identification equations. On the other hand, the hybrid representation has a higher risk of producing underdetermined identification equations because too many uncertainty parameters appear in too few equations.

This is in fact what happens in the 2D case, if the twists and wrenches are measured by a "classical" robotic system: a serial-chain robot arm holds the *MO*; the motion of the *MO* is derived from the end effector motion of the robot; the wrench exerted on the *MO* is derived from the force measured by the robot's wrist force/torque sensor. For this kind of system, the inertial and body-fixed representations each involve only one of the two uncertaintiesΔq^+ and $\Delta q'^+$ (because they each need only one reference frame, $\{bs\}$ and $\{ee\}$ respectively, to express the motion of the *MO*); however, both uncertainties occur in the hybrid representation (because this representation uses both $\{bs\}$ and $\{ee\}$), which results in underdetermined equations. The following paragraphs illustrate these statements more clearly with concrete examples.

6.7.1 First order twist based identification

The twist based reciprocity relation, Eq. (6.12), and the series expansion of the wrench Jacobian, in the hybrid representation and in terms of Δq^+ and $\Delta q'^+$, Eqs. (B.6)-(B.7), give:

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}^T \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \Delta q^+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -r^+ \\ 0 \end{bmatrix} + \Delta q'^+ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\} = 0. \tag{6.22}$$

The inertial representation, using the series expansion of Eq. (B.12), yields:

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}^T \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \Delta q'^{+} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -r'^{+} \\ 0 \end{bmatrix} \right\} = 0. \tag{6.23}$$

Finally, the body-fixed representation, using the series expansion of Eq. (B.17), yields:

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}^T \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \Delta q^{+} \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -r^{+} \\ 0 \end{bmatrix} \right\} = 0. \tag{6.24}$$

Remember that the same formal twist components ($v_x \ldots \omega_z$) have different values in each of the equations (6.22)-(6.24)! Rewriting these equations results in:

1. One equation for the two uncertainties in the hybrid representation:

$$v_z - r^{+}\omega_y \Delta q^{+} + v_x \Delta q'^{+} = 0. \tag{6.25}$$

   In general, this equation is not solvable. Exceptions occur when either v_x or ω_y are zero.

2. One equation in the single uncertainty $\Delta q'^{+}$ for the inertial representation:

$$v_z + \Delta q'^{+}(v_x - r'^{+}\omega_y) = 0. \tag{6.26}$$

3. One equation in the single uncertainty Δq^{+} for the body-fixed representation:

$$v_z - \Delta q^{+}(v_x + r^{+}\omega_y) = 0. \tag{6.27}$$

The reason for these differences is clarified by FIG. 6.4. The modelled contact frames on *MO* and *ENV* —i.e., the virtual manipulator's base frame $\{bs\}$ and end effector frame $\{ee\}$, respectively— are connected by two kinematic chains: one contains the robot; the other one is the virtual uncertainty manipulator between $\{bs\}$ and $\{ee\}$, parameterized by the unknown parameters Δq^{+} and $\Delta q'^{+}$. These uncertainties are caused by the inaccurate knowledge of where exactly the contact point lies on the circles representing the *MO* and the *ENV*. Since the positions of the end effector frame $\{ee\}$ and the base frame $\{bs\}$ are modelled with respect to the robot, a measurement of the robot's end effector twist yields the twist of the $\{ee\}$ and $\{bs\}$ frames after a simple screw transformation. In order to move

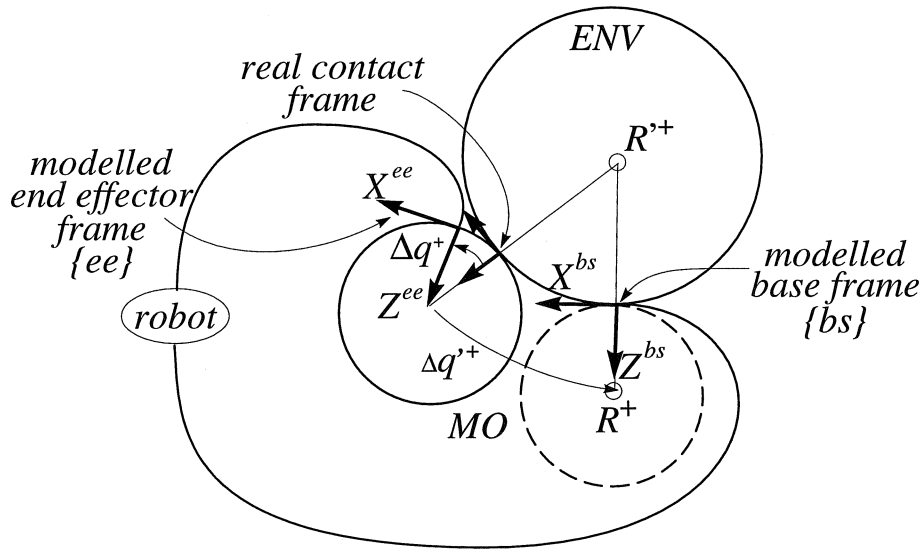


FIGURE 6.4 : *Kinematic loop of 2D contour following.* The locations of the modelled contact frames —i.e., the virtual manipulator's base and end effector frames, respectively— are linked by: 1) the kinematic uncertainty chain, with the unknown parameters Δq^+ and $\Delta q'^+$; 2) the manipulating robot that moves the *MO* over the *ENV*.

the *MO* in a desired way over the *ENV*, it suffices to know the uncertainty Δq^+ only. This can be achieved by the body-fixed identification equation (6.27), since this representation needs only the $\{ee\}$ frame as a reference. The inertial representation needs only the $\{bs\}$ frame, and the hybrid representation needs both of them. Hence, in this last case, the measured robot twist is to be transformed over both uncertainties, and this causes the underdetermined equation (6.25).

The following paragraphs give geometrical interpretations for the inertial and body-fixed representations; the hybrid representation is some sort of "coupled mixture" of both other representations, and this fact complicates the geometrical interpretation.

The identification equation (6.27) in the body-fixed representation is easily interpreted, see the left-hand side of FIG. 6.5:

1. Let the measured robot twist transformed to the modelled end effector frame $\{ee\}$ be given by $[0\ \omega_y\ 0\ v_x\ 0\ v_z]^T$.

2. Transform this twist to a parallel frame, with origin in the centre of curvature of the *MO*. This yields the twist $[0\ \omega_y\ 0\ v_x + r^+\omega_y\ 0\ v_z]^T$.

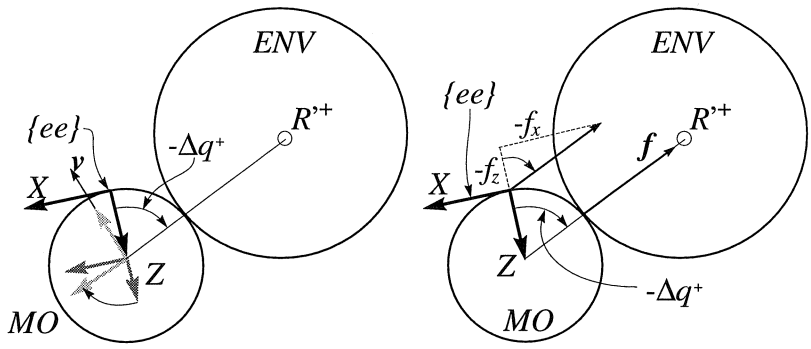


FIGURE 6.5 : *2D contour following : first order identification of* Δq^+. Left: transformation of the modelled contact frame $\{ee\}$ to the light gray frame at the centre of the MO whose Z axis is aligned with the real contact normal. In this frame, the Z component of the translational velocity three-vector vanishes. The medium gray frame is an intermediate frame used for the geometrical interpretation of the Eq. (6.27). Right: components of the real contact force $\boldsymbol{f}$ along the axes of the modelled contact frame $\{ee\}$.

3. Rotate this twist over the uncertainty Δq^+. The approximations $\cos(\Delta q^+) \approx 1$ and $\sin(\Delta q^+) \approx \Delta q^+$ give the rotation matrix:

$$\boldsymbol{R} \approx \begin{bmatrix} 1 & 0 & -\Delta q^+ \\ 0 & 1 & 0 \\ \Delta q^+ & 0 & 1 \end{bmatrix}.$$

   Hence, the transformed twist is $[0\ \omega_y\ 0\ (v_x + r^+\omega_y - \Delta q^+ v_z)\ 0\ (v_z + \Delta q^+(v_x + r^+\omega_y))]^T$.

4. Since the centre of the MO can only move in the tangential direction, the Z component of its translational velocity three-vector must vanish. This is exactly the identification equation (6.27).

The identification equation (6.26) for $\Delta q'^+$, in the inertial representation is derived in a completely similar way, with $\{bs\}$ replacing $\{ee\}$, FIG. 6.6.

Remark A pure rotation about the centre of curvature of the MO makes $v_x + r^+\omega_y$ zero. Hence, the uncertainty Δq^+ becomes unidentifiable by Eq. (6.27). This is easily understood, since this rotation is always possible regardless of where the ENV contacts the MO.

Remark Transforming the identification equation (6.27) to the frame at the centre of the *MO* yields:

$$\Delta q^+ = -\frac{v_z}{v_x}. \tag{6.28}$$

This is exactly the identification equation used for the 2D *vertex-surface* contact type, FIG. 2.1 and Section 6.8.1!

Remark The identification equation (6.27) can also be used to estimate the instantaneous radius of curvature r'^+ of the *ENV*. An important necessary condition is that there is no slip, i.e., tracking keeps the *MO* in a constant orientation with respect to the contact normal. So, $\dot{q}'^+ = \omega_y$, and:

$$\Delta q^+ = \frac{v_z}{v_x} \quad \overset{(Fig.\ 6.7)}{=} \quad \frac{v\sin(\Delta q^+)}{v_x} \approx \frac{(r'^+\omega_y)(\Delta q^+)}{v_x} \quad \Rightarrow \quad r'^+ = \frac{v_x}{\omega_y}. \tag{6.29}$$

This result corresponds to the classical formula expressing the tangential velocity v_x of a point at a distance r'^+ of a rotation centre with angular velocity ω_y. Remark however that: 1) the Z axis of the frame in which the measurements are interpreted need not be normal to the surface, and 2) the actual misalignment need not be known, as long as it remains constant and small.

6.7.2 First order wrench based identification

From a mathematical point of view, wrench based identification is not very different from twist based identification. The same remarks concerning the three twist and wrench representations apply.

Hybrid representation The wrench based reciprocity relation, Eq. (6.13), and the series expansion of the twist Jacobian, in the hybrid representation and in terms of Δq^+ and $\Delta q'^+$, Eqs. (B.2) and (B.8), give:

$$0 = \begin{bmatrix} m_x \\ m_y \\ m_z \\ f_x \\ f_y \\ f_z \end{bmatrix}^T \left\{ \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ r'^+ & -r^+ \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + \Delta q^+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ r^+ & r^+ \end{bmatrix} + \Delta q'^+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -r'^+ & r^+ \end{bmatrix} \right\}. \tag{6.30}$$

This yields two equations in Δq^+ and $\Delta q'^+$:

$$\begin{cases} 0 = m_y + r'^+ f_x + \Delta q^+ f_z r^+ - \Delta q'^+ f_z r'^+, \\ 0 = m_y - r^+ f_x + \Delta q^+ f_z r^+ + \Delta q'^+ f_z r^+. \end{cases}$$

Rewriting gives:

$$\Delta q^+ = -\frac{m_y}{r^+ f_z}, \quad \Delta q'^+ = \frac{f_x}{f_z}. \tag{6.31}$$

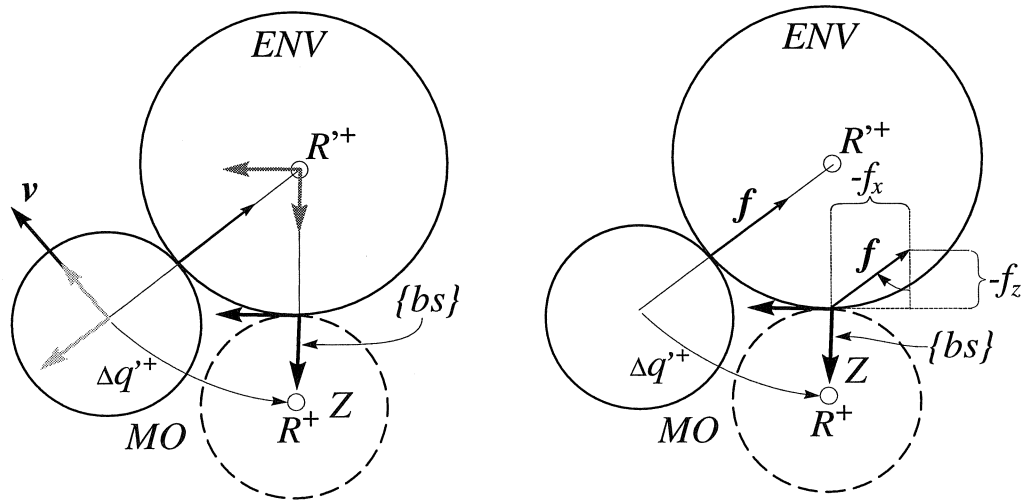


FIGURE 6.6 : *2D contour following: first order identification of* $\Delta q'^{+}$. Left: transformation of the modelled contact frame $\{bs\}$ to the light gray frame at the centre of the MO whose Z axis is aligned with the contact normal. In this frame, the Z component of the translational velocity three-vector vanishes. The medium gray frame is an intermediate frame used for the geometrical interpretation of Eq. (6.26). Right: the components of the real contact force $\boldsymbol{f}$ along the axes of the modelled contact frame $\{bs\}$.

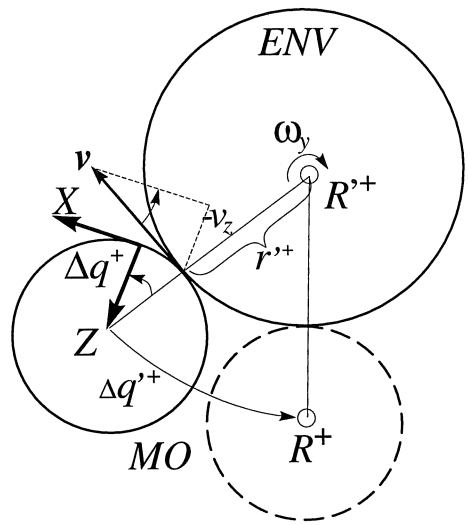


FIGURE 6.7 : *2D contour following: normal velocity in a no-slip motion.*

Inertial representation The inertial representation uses the derivative of the wrench Jacobian as given by Eq. (B.11), which results in:

$$\begin{cases} 0 &= m_y + r'^{+} f_x, \\ 0 &= m_y - r^{+} f_x + \Delta q'^{+} f_z (r^{+} + r'^{+}). \end{cases}$$

Elimination of m_y yields:

$$\Delta q'^{+} = \frac{f_x}{f_z}. \tag{6.32}$$

FIGURE 6.6 clarifies this identification equation, when making the approximation $\tan(\Delta q'^{+}) \approx \Delta q'^{+}$.

Body-fixed representation In a completely similar way one gets:

$$\Delta q^{+} = -\frac{f_x}{f_z}. \tag{6.33}$$

Again, this is easily interpreted geometrically, FIG. 6.5, and corresponds to the 2D *vertex-surface* case.

6.7.3 Second order twist based identification

The second order wrench based identification equation, Eq. (6.14), uses Eq. (3.41) that gives the time derivative of a Jacobian matrix in function of the partial derivatives with respect to the joint angles. Again, remark that the numerical values of the measurement components have different meanings in the three representations!

Hybrid representation The hybrid representation uses the series expansion of the wrench Jacobian, Eq. (B.6). This gives:

$$0 = \begin{bmatrix} a_x \\ a_y \\ a_z \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}^T \left\{ \begin{bmatrix} \dot{q}'^{+} \\ 0 \\ 0 \\ 0 \\ -r^{+}\dot{q}^{+} \\ 0 \end{bmatrix} - \Delta r^{+} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dot{q}^{+} \\ 0 \end{bmatrix} + \Delta r'^{+} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\} \tag{6.34}$$

$$\Rightarrow \quad 0 = a_z - \dot{q}^{+}\omega_y (r^{+} + \Delta r^{+}) + \dot{q}'^{+} v_x. \tag{6.35}$$

Inertial representation The inertial representation uses the series expansion Eq. (B.11). This gives:

$$r'^{+} + \Delta r'^{+} = \frac{a_z + \dot{q}'^{+} v_x}{\dot{q}'^{+} \omega_y}. \tag{6.36}$$

Interpretation of this identification equation is rather simple for the case of no slip, i.e., $\dot{q}^+ = 0, \dot{q}'^+ = \omega_y$:

1. The contact frame rotates with a constant velocity ω_y around the R'^+ joint at a nominal distance of r'^+, but in reality at a distance of $r'^+ + \Delta r'^+$. Hence, one measures an acceleration error a_z between the modelled and real contact frames.

2. The *model* detects no slip because v_x and ω_y are compatible with the nominal radius r'^+, i.e., $v_x = r'^+\omega_y$.

3. Hence, the velocity of R'^+ is estimated as $\dot{q}'^+ = \omega_y$, and the estimated $\dot{q}^+$ is zero.

4. Thus, the identification equation (6.36) reduces to:

$$\frac{\Delta r'^+}{r'^+} = \frac{a_z}{v_x \omega_y}. \tag{6.37}$$

   This means that the relative error in the real curvature radius is estimated as the ratio between the measured acceleration error and the modelled normal acceleration.

Body-fixed representation The body-fixed representation uses the series expansion Eq. (B.17). A similar reasoning as for the inertial representation gives:

$$r^+ + \Delta r^+ = \frac{a_z - \dot{q}^+ v_x}{\dot{q}^+ \omega_y}, \tag{6.38}$$

with a similar geometric interpretation.

6.7.4 Second order wrench based identification

Because the time derivatives of wrenches don't have any well-known interpretations similar to the twist rate expression $a_z = v_x\omega_y = r(\omega_y)^2$ for motion on a circle, second order wrench based identification yields rather involved equations, for which no easy geometric interpretation can be given. For example, in the hybrid representation, Eqs. (5.7), (3.41) and the wrench based version of Eq. (6.14), give:

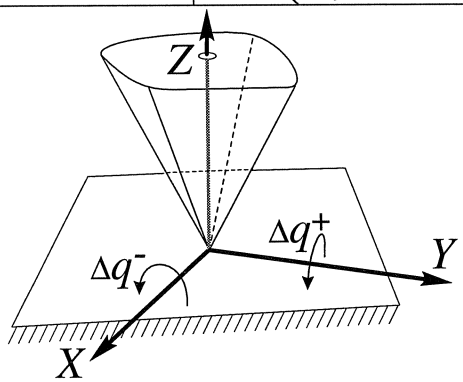
$$\begin{aligned} 0 = & \begin{bmatrix} \dot{m}_x \\ \dot{m}_y \\ \dot{m}_z \\ \dot{f}_x \\ \dot{f}_y \\ \dot{f}_z \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ r'^+ & -r^+ \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} m_x \\ m_y \\ m_z \\ f_x \\ f_y \\ f_z \end{bmatrix}^T \left\{ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dot{q}^+ r^+ - \dot{q}'^+ r'^+ & (\dot{q}^+ + \dot{q}'^+)\, r^+ \end{bmatrix} \right. \\ & \left. + \Delta r \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dot{q}^+ & \dot{q}^+ + \dot{q}'^+ \end{bmatrix} + \Delta r' \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -\dot{q}'^+ \end{bmatrix} \right\}. \end{aligned} \tag{6.39}$$

6.8 First order identification of polyhedral contacts

This Section treats the very common contact situations in which both *MO* and *ENV* are polyhedral, i.e., either a *vertex*, a straight *edge* or a planar *face*. The *vertex-vertex*, *vertex-edge* and *edge-vertex* contact situations are not stable and hence are not treated. Moreover, only time varying uncertainties are assumed, so the *face-face* contact is not dealt with either. Five polyhedral contact situations remain: *vertex-face*, *edge-face*, *edge-edge* (edges are not parallel!), *face-vertex* and *face-edge*. These polyhedral contacts can all be specified and executed by the Task Frame Formalism (TFF), see Chapter 1. The body-fixed representation is chosen to represent twists and wrenches, since this most closely corresponds to the TFF approach: the motion of the *MO*, as well as the contact forces exerted on the *MO*, are specified and measured as the motion of, and the forces on, the task frame, *and* expressed in this same task frame. The calculations are straightforward, along the lines developed in Sections 6.3 and 6.4, and with the Jacobian derivatives as in Chapter 3 (taking the appropriate limits for the polyhedral geometry!). Only the results (including the monitoring equations) are listed below; they turn out to be easily interpretable by means of the accompanying figures. It is assumed that the position and orientation of the *MO* with respect to the robot end effector are known. Hence, the uncertainties stem from inaccurate knowledge of the *ENV* only. It is straightforward to incorporate uncertainties in the *MO* too.

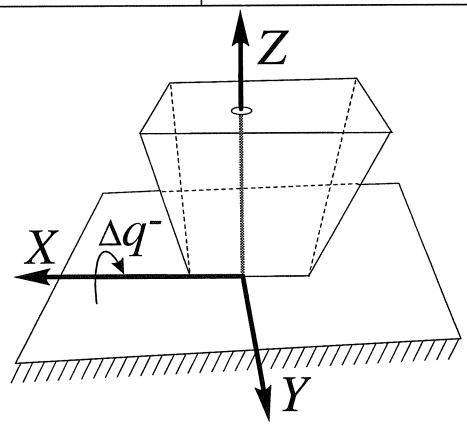
6.8.1 Vertex-face

The uncertainties are Δq^+ and Δq^-, corresponding to rotations about the Y and X axes, respectively.

twist based	wrench based
identification	
$v_z + \Delta q^+ v_x - \Delta q^- v_y = 0$	$\begin{cases} f_x - \Delta q^+ f_z = 0 \\ f_y + \Delta q^- f_z = 0 \end{cases}$
	
monitoring	
$v_z = 0$	$f_x = f_y = m_x = m_y = m_z = 0$

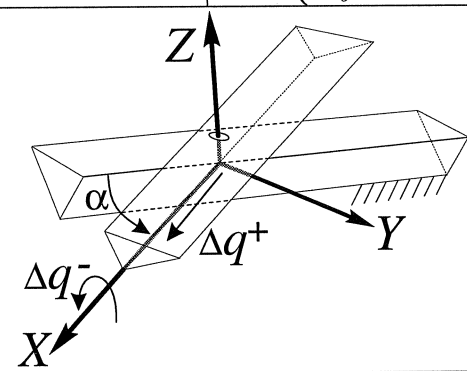
6.8.2 Edge-face

The only uncertainty is Δq^-, corresponding to a rotation about the X axis.

twist based	wrench based
identification	
$\begin{cases} v_z - \Delta q^- v_y = 0 \\ \omega_y + \Delta q^- \omega_z \end{cases}$	$f_y + \Delta q^- f_z = 0$
	
monitoring	
$v_z = \omega_y = 0$	$f_x = f_y = m_x = m_z = 0$

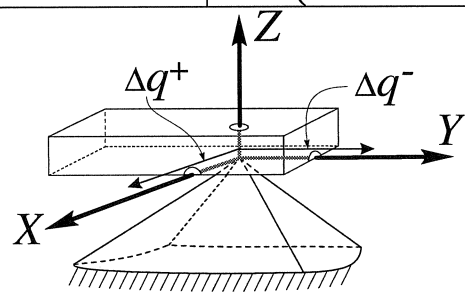
6.8.3 Edge-edge

The uncertainties are Δq^- and Δq^+, corresponding to a rotation about and a translation along the X axis, respectively. The uncertainty α in the relative directions of the edges is unidentifiable.

twist based	wrench based
identification	
$v_z - \Delta q^- v_y - \Delta q^+ \omega_y = 0$	$\begin{cases} m_y + \Delta q^+ f_z = 0 \\ f_y + \Delta q^- f_z = 0 \end{cases}$
	
monitoring	
$v_z = 0$	$f_x = f_y = m_x = m_y = m_z = 0$

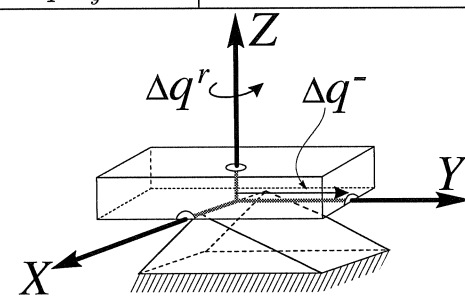
6.8.4 Face-vertex

The uncertainties are Δq^+ and Δq^-, corresponding to translations along the X and Y axes, respectively.

twist based	wrench based
identification	
$v_z + \Delta q^- \omega_x - \Delta q^+ \omega_y = 0$	$\begin{cases} m_y + \Delta q^+ f_z = 0 \\ m_x - \Delta q^- f_z = 0 \end{cases}$
	
monitoring	
$v_z = 0$	$f_x = f_y = m_x = m_y = m_z = 0$

6.8.5 Face-edge

The uncertainties are Δq^- and Δq^r, corresponding to a translation along the Y axis, and a rotation about the Z axis, respectively.

twist based	wrench based
identification	
$\begin{cases} v_z - \Delta q^- \omega_x = 0 \\ v_z + \Delta q^r \omega_y = 0 \end{cases}$	$m_x - \Delta q^r m_y - \Delta q^- f_z = 0$
	
monitoring	
$v_z = \omega_y = 0$	$f_x = f_y = m_x = m_z = 0$

6.9 Conclusions

This Chapter used the Taylor series expansions of the Jacobian matrices in terms of the uncertainty parameters to derive identification equations for an arbitrary contact situation

and an arbitrary set of uncertainty parameters. Reciprocity and consistency have been applied as physical "first principles" to restrict the uncertainty search space. Although both approaches start from very different systems of equations linking the geometric uncertainties to the force and motion measurements, they eventually lead to exactly the same results. Depending on whether the force or motion measurements are used as inputs, the identification equations are called wrench based and twist based, respectively. Identification equations have been derived for first order as well as second order geometric uncertainties: the position of the contact point, the direction of the contact normal, and the curvature parameters in the neighbourhood of the contact point. The general identification equations have been applied to the example of the 2D contour following task, as well as to the most common polyhedral contact tasks.

The consistency based identification equations are more complicated to solve, because of their non linearity. However, as a by-product, they lead to a number of monitoring equations, that can be used to detect a drastic change in the current contact situations. Hence, these monitoring equations allow to find out whether a transition to another contact type has occured.

The following guidelines help to practically use the identification equations derived in this Chapter:

1. Try to find a minimum set of uncertainty parameters for the expected contact type(s). In other words: try to use as much as possible a priori information about the contact.

2. Try to use a reference frame in which the representation of the uncertainties is as simple as possible.

3. Define for each uncertainty whether it is time varying or time invariant.

4. Use either identification approach to derive a set of identification (and monitoring) equations for the specified uncertainties.

5. If the resulting set of equations is overdetermined, select the equations which are best conditioned (depending on friction, compliance, and other disturbances in the compliant motion system).

6. Time invariant uncertainties could be identified by some sort of batch identification algorithm.

7. If the set of identification equations is underdetermined, one should apply active sensing actions in order to fully excite all uncertainty parameters.

8. Design a good tracking feedback loop; i.e., its dynamics should be adapted to the sensing and control modules.

These guidelines lead to a potentially autonomous compliant motion system. However, many aspects of a fully autonomous system are still lacking in the current state-of-the-art

in robotics: intelligent and robust planning of active sensing, automatic selection of the best conditioned equations, robust identifiation approaches for underdetermined identification, robust sensor signal processing, ... Moreover, for every implementation, the eventual robustness of the total compliant motion system to the geometric motion constraint uncertainties occurring in a given task cannot be predicted, but only assessed experimentally.

Chapter 7

Experiments and applications

This Chapter illustrates the theoretical developments of the previous Chapters by means of real world experiments. Firstly, Section 7.1 briefly describes a typical hardware and software setup used for the experiments. Then, Section 7.2 demonstrates motion specification with a virtual manipulator by means of the "sphere-on-sphere" contact situation. "Peg-on-hole," Section 7.3, and "incompatible 3D seam following," Section 7.4, illustrate model based motion specification with multiple and time varying contacts. The "2D contour following" task, Section 7.5, shows some results in first and second order identification, as well as the limits of the current compliant motion control system. Section 7.6 describes how the kinematic approach to constrained motion can be extended to "virtual contacts," with an application in redundancy resolution. Finally, Section 7.7 discusses two other application areas for which no implementation exists yet: simulation of compliant motion with curved surfaces, and calibration of robot kinematics using force control.

The robotics literature contains no reports on compliant motion tasks with the same level of complexity as most of the tasks described in this Chapter![1]

7.1 Controller hardware and software

The hardware setup is a shown in FIG. 7.1: the robot arm is an industrial KUKA-IR361 robot, equipped with a SCHUNK wrist force-torque sensor and a compliant element made in-house, to which the *MO* is rigidly fixed. This compliance is a necessary component of the force control approach used in the robot controller. This controller works with an internal analog velocity control loop, around which an "external" force control loop is implemented. Set-point and tracking control are not the topics of this text, hence the interested reader should consult De Schutter (1986), De Schutter & Van Brussel (1988b), and Van de Poel et al. (1993) for details of this force controller. The tracking loop simply consists of proportional feedback: the identified uncertainty is transformed to a proportional velocity set-point.

[1] With the exception of publications from the same research group, e.g., Demey and De Schutter (1994) on 2D contour following with on-line identification of curvature.

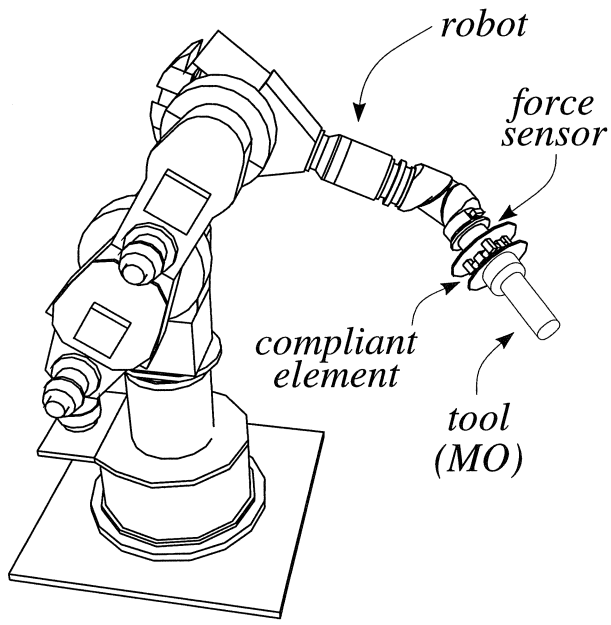


FIGURE 7.1 : *Hardware setup used in the experiments.* It consists of a velocity controlled robot, a stiff force sensor, a compliant element, and finally the *MO*. The motion of the *MO* is derived from the measured positions of the robot's joints. (Figure courtesy of Bagus Arthaya.)

Maybe the easiest way to understand the basics of this external force control, is to look at it as a "deformation controller": an error between the desired force and the measured force is transformed into a velocity of the end effector that tends to deform the compliant element in such a way that the force error will decrease. The major disadvantage of this compliant element —at least for the purpose of motion constraint identification— is that it introduces significant disturbances in the "measurement" of the twist executed by the *MO*. To cope with this inconvenience, a compliant force sensor has been developed, Degezelle & Peeters (1994), which measures its own deformations. In this way, the "disturbances" introduced by the deformation of the compliant element can be compensated. However, at the time of writing, this new sensor's software was not yet available in a sufficiently workable form.

The motion of the *MO* is derived from the motion of the robot joints, based on the nominal kinematics of the robot and the screw transformations between the robot end effector frame and the contact frame on the *MO*. Only the joint *positions* are effectively measured, by means of digital encoders. Hence, the *MO*'s "measured" velocities are disturbed by discretisation errors. This is even more the case for the accelerations.

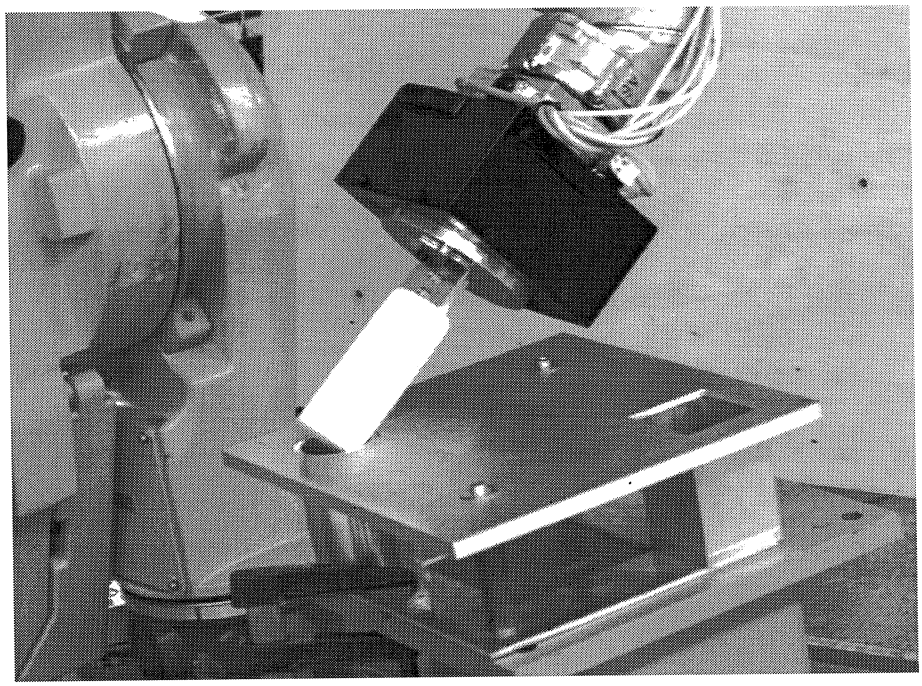


FIGURE 7.2 : *Peg-on-hole: experimental setup.*
(Figure courtesy of S. Demey and Adhi Sudadi.)

7.2 Sphere-on-sphere

This task illustrates Chapter 4, i.e., given a good model of the motion constraint, motion specification generates robot control set-points that move the *MO* with only little tracking error.

"Sphere-on sphere" is the most "pure" application of the kinematic modelling approach as presented in Chapter 4: the *MO* and the *ENV* are both spheres. Hence, the kinematic model of the contact is exactly as depicted in FIGS. 4.2 and 4.3. The task is specified by applying desired velocities to one or more of the five joints in the kinematic model. The resulting open loop motion of the *MO* corresponds to the modelled motion within the absolute accuracy range of the robot. No plots are given for the (successful) real world experiments with this task: indeed, showing a plot with a series of consecutive positions of the *MO* sphere moving over the *ENV* sphere rapidly becomes very confusing, to draw as well as to interpret. Anyway, integration of this "sphere-on-sphere" task in a classical Task Frame controller is rather straightforward using the Jacobian expressions of Appendix A.

This experiment proves also that a simple "first order" motion specification —i.e., the desired joint velocities of the virtual kinematic chain— generates a good second order motion for the *MO*, in the sense that no feedback is necessary to keep the contact forces within acceptable limits, even for large amplitude motions. This assumes of course that the system has a (small) amount of passive compliance, and the geometry of the kinematic model is rather accurate.

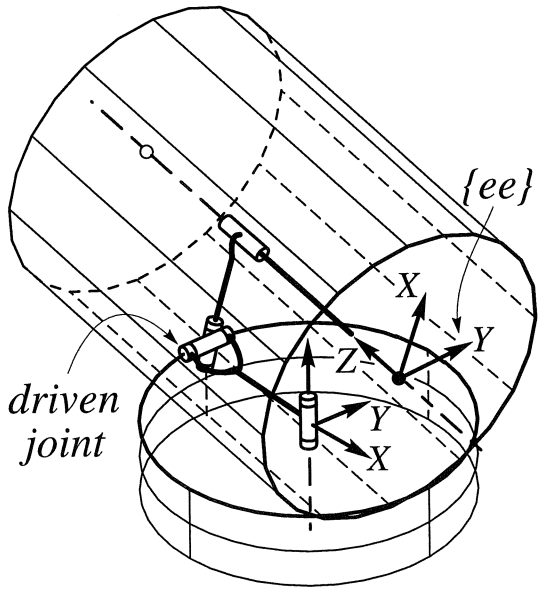


FIGURE 7.3 : *Peg-on-hole: driven joint.*

7.3 Peg-on-hole

This task, FIG. 7.2, also illustrates the developments of Chapter 4. Its modelling and motion specification have already been described in detail in Section 4.4.3. The alignment of the peg on the hole (both have a radius of $25mm$) is realized by specifying a velocity for the alignment joint, FIG. 7.3. The motion set-points are derived from the nominal velocity profiles given in FIG. 4.9: the driving alignment velocity is $-0.2rad/sec$, and the initial alignment error is about $60deg$. The task is executed without force feedback. The inevitable errors between model and reality are absorbed by the compliant element. Its stiffness matrix $[k_x\ k_y\ k_z\ k_{xx}\ k_{yy}\ k_{zz}]^T$ —with respect to a frame parallel to $\{ee\}$, FIG. 7.3, at its centre of compliance— is approximately [45N/mm 60N/mm 45N/mm 175000Nmm/rad 225000Nmm/rad 175000Nmm/rad]^T. FIGURE 7.4 shows the deformation energy stored in the compliant element during the alignment motion:

$$E_d = k_y d_y^2 + k_z d_z^2 + k_{xx}\delta_x^2,$$

with d_y and d_z the translational deformations (mm) in Y and Z, and δ_x the angular deformation (rad) about X with respect to the frame at the centre of compliance.

The alignment motion is programmed with a *constant* angular velocity $\dot{\alpha}$. Hence, an acceleration jump occurs in the motion of the robot. Due to inertial effects, the peg does not immediately follow the motion of the robot end effector, and hence the deformation energy shows a peak at the start of the motion. After this inertial transition phase, the deformation energy remains more or less constant, indicating that the discrepancies between model and reality remain small. Visual inspection reveals also that the contacts are perfectly maintained.

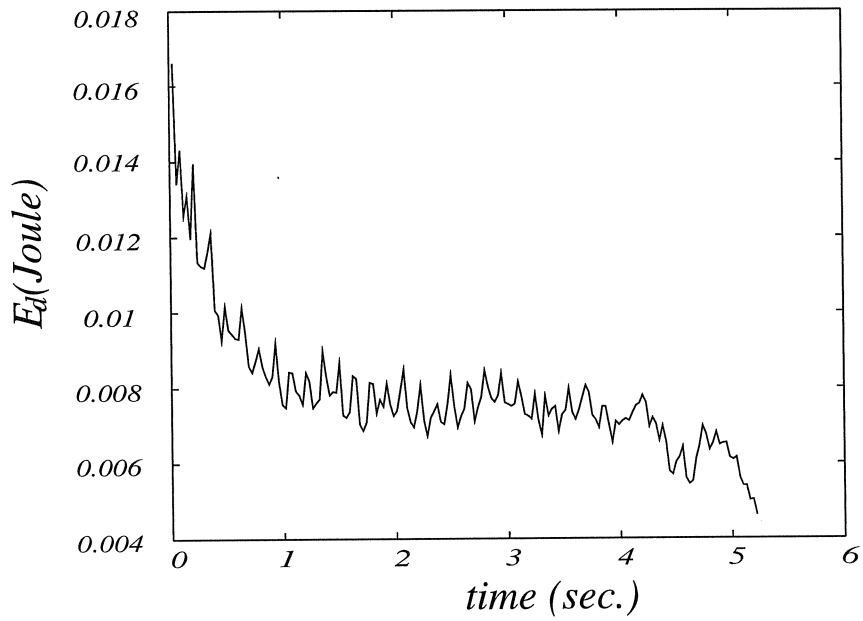


FIGURE 7.4 : *Peg-on-hole: deformation energy of the compliant structure during alignment.* The initial alignment error was about $60deg$, the specified angular alignment speed was: $\dot{\alpha} = -0.2\ rad/sec.$ (Figure courtesy of S. Dutré.)

At the end of the alignment task, the axis of the peg is almost perfectly in line with the axis of the hole. The compliant element can almost completely free its potential energy, see FIG. 7.4. The subsequent insertion of the peg succeeds even *without active force control.* The clearance between peg and hole is about $\pm.25mm$; peg and hole are both chamferless.

The success rate of this experiment is high, once the three point contact situation is reached. This proves the accuracy of the model. Sometimes, however, the alignment motion fails in the sense that the insertion cannot take place because the peg is not exactly above the hole. The reason for this failure is not due to the modelling or the motion specification theory, but to the fact that the velocity profile of FIG. 4.9 implies jumps in the acceleration at the start and the end of the motion. This excites some of the resonance frequencies of the compliant element, such that the peg sometimes deviates from the correctly aligned position. Moreover, in the aligned position the contact situation is in some sort of degenerate state, in which the contacts are not very stable and provide little damping to the oscillations of the compliant element. Hence, it would be better to avoid these oscillations by using a smooth acceleration and deceleration phase for the driving joint velocity.

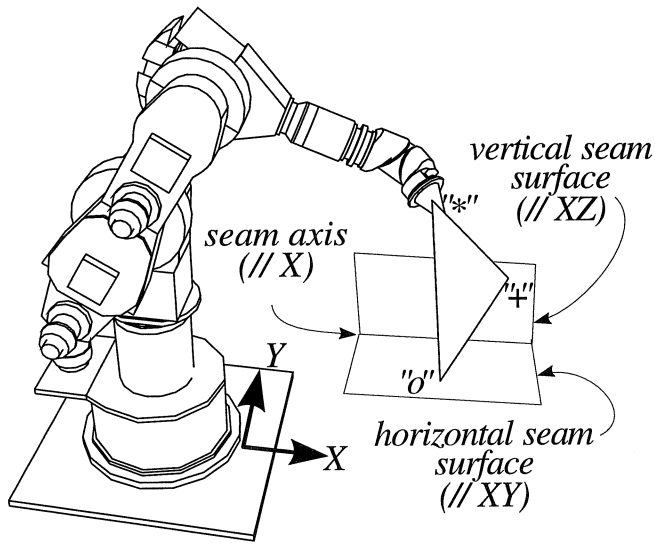


FIGURE 7.5 : *3D seam following: experimental setup.* The seam axis is horizontal, along the X axis of the manipulator base frame. One seam surface is parallel with the manipulator's XY plane; the other seam surface is parallel to the XZ plane. The MO is a triangle; its vertices are labelled "o," "+" and "$*$."

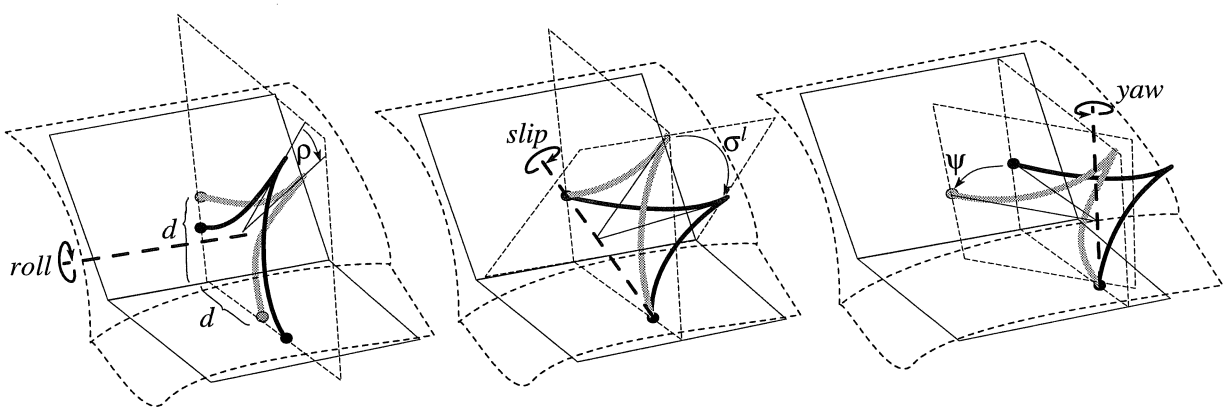


FIGURE 7.6 : *3D seam following: driving degrees-of-freedom.* The following figures show the measured motion of the MO generated by specifying a sinusoidally varying driving velocity to, respectively, the *slip*, *slide*, *yaw* and *roll* degrees-of-freedom. *Slide* is simply a motion along the seam; the other degrees-of-freedom are shown above.

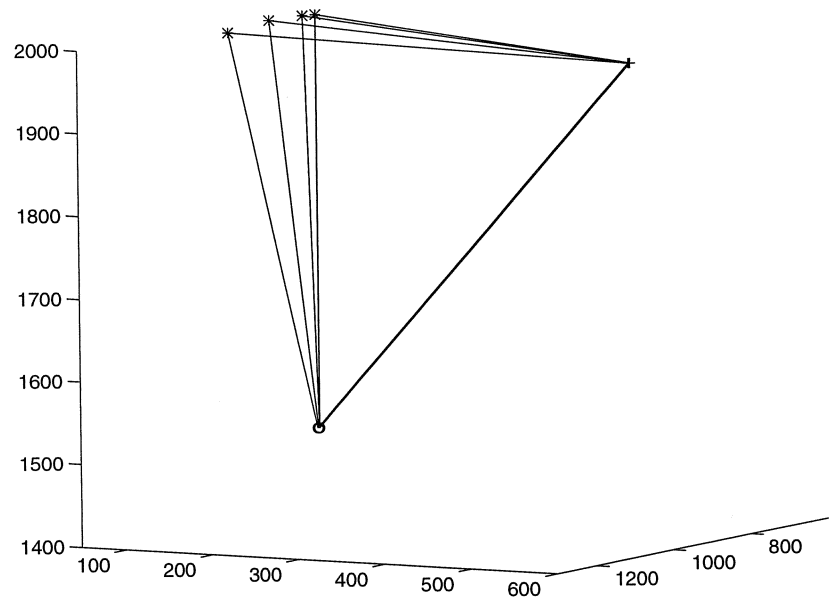


FIGURE 7.7 : *3D seam following: slip motion.* Both contact points, indicated by the "o" and "+" signs, remain motionless, while the *MO* rotates about the line between these two points.

7.4 Incompatible 3D seam following

As the previous two tasks, this task illustrates model based motion specification. However, this time *two* contacts constrain the motion of the *MO*. Indeed, generalizing 2D contour following to three dimensions involves controlled motion of a tool that has two point contacts with its environment, see FIG. 7.5 and App. E. In fact, this task is equivalent to two surface following tasks in parallel. In general, the tool's geometry is incompatible with the seam formed by the two environment surfaces. In this experiment, these surfaces are planar and make an angle of $90deg$. Hence, two *vertex-surface* contacts occur, which constrain the tool to four degrees-of-freedom. Appendix E describes the mathematical details of how to model the motion freedom of this constraint. It also derives an intuitive way to specify the motion of the *MO* in terms of *slide, slip, roll,* and *yaw* velocities, FIG. 7.6.

As for the "sphere-on-sphere" task, the experiment again applies independent driving velocity set-points to each of these degrees-of-freedom. FIGURES 7.7-7.11 plot the measured open loop motions (i.e., without real contacts) of a triangular *MO*, with, respectively, slip, slide, yaw and roll velocity inputs, FIG. 7.6. These velocity set-points for all degrees-of-freedom are sine waves, with amplitudes of $0.25rad/sec$ and a frequency of $0.1Hz$. In all plots, the vertices indicated by "o" and "+" are the contact points, FIG. 7.5; the "$*$" vertex is a third reference point on the *MO*. It is clear from these plots that the contacts are obeyed, and, thus, that the kinematic approach offers a faithful model of this complex time varying motion constraint.

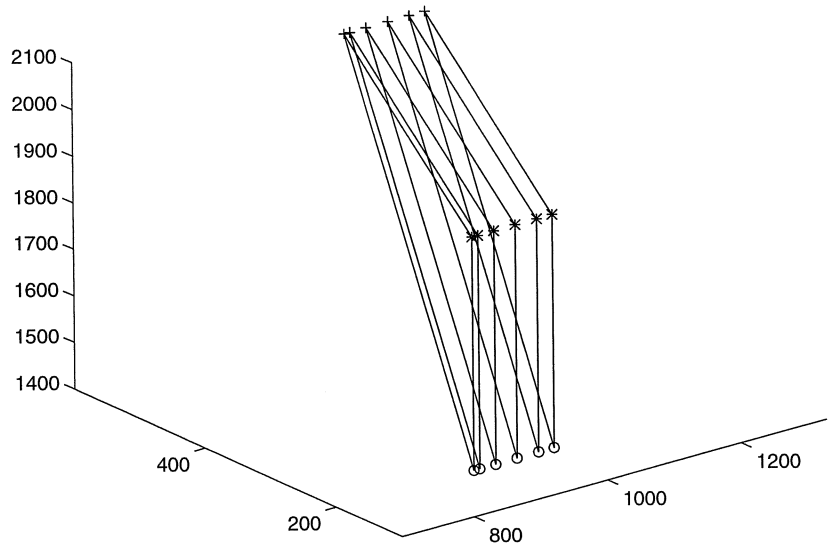


FIGURE 7.8 : *3D seam following: slide motion.* The *MO* moves parallel to the seam axis.

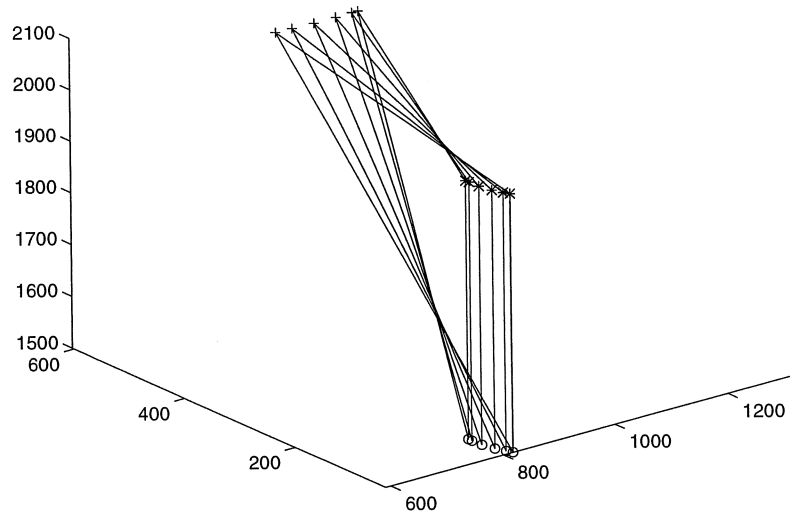


FIGURE 7.9 : *3D seam following: yaw motion.* This 3D perspective can be a bit confusing; FIG. 7.10 shows the motion from two other viewpoints.

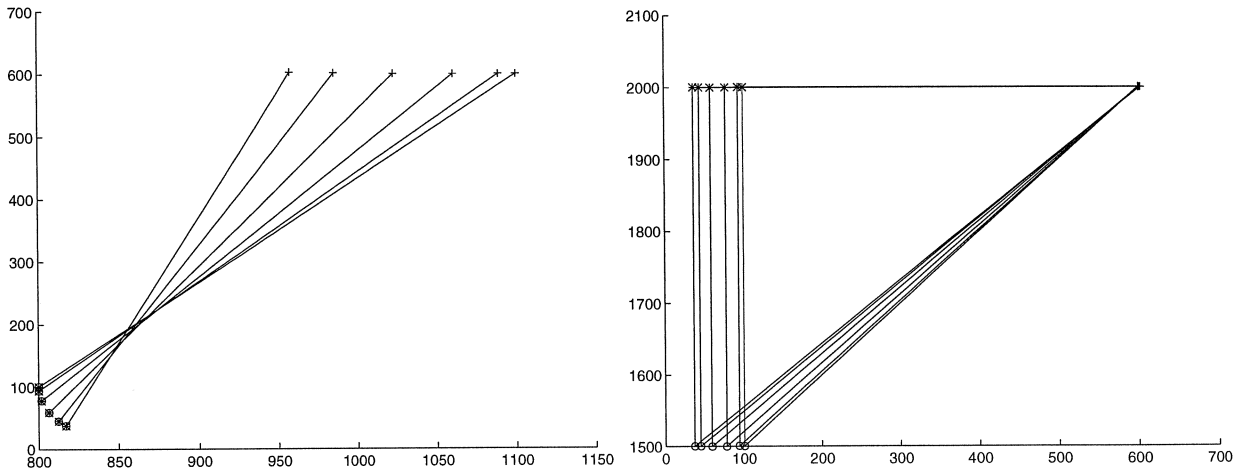


FIGURE 7.10 : *3D seam following: yaw motion, from other viewpoints.* The left-hand side plot shows the same motion as in FIG. 7.9, but as seen from above: it proves that the "+" vertex nicely follows the vertical seam surface. The right-hand side plot shows the view along the seam axis: it proves that also the second vertex, "o," obeys its local constraint.

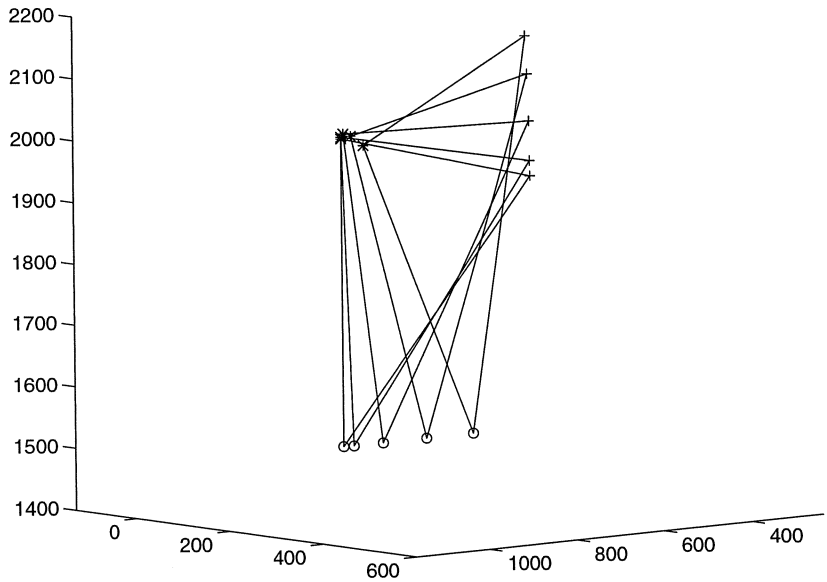


FIGURE 7.11 : *3D seam following: roll motion.* This 3D perspective can be a bit confusing; FIG. 7.12 shows the motion from two other viewpoints.

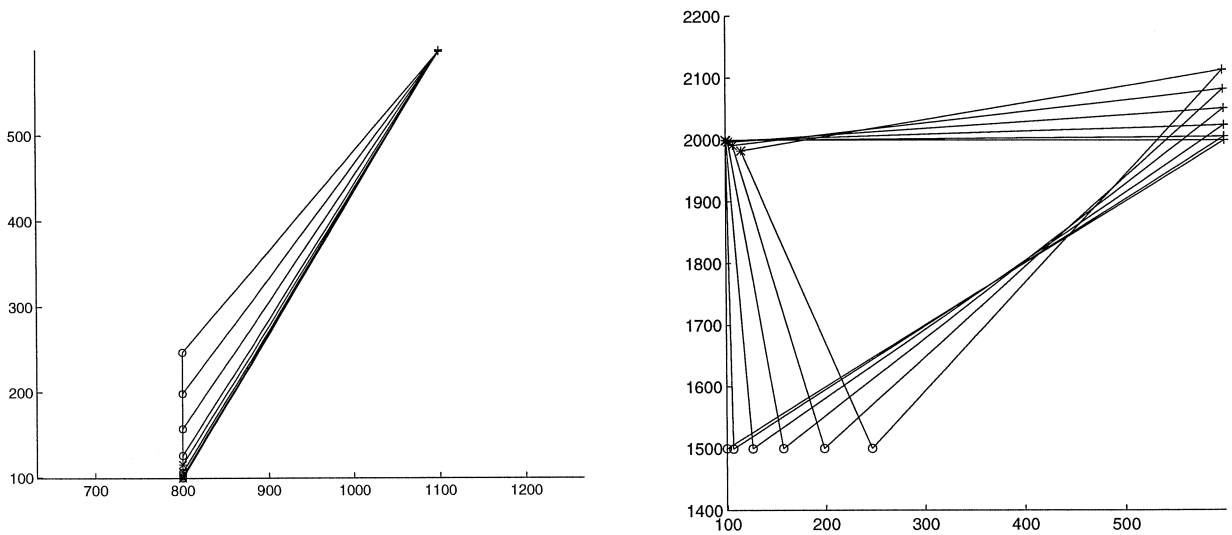


FIGURE 7.12 : *3D seam following: roll motion, from other viewpoints.* Left: seen from above, and with the seam axis parallel to the horizontal axis of the figure. It proves that both "+" and "o" vertices move orthogonal to the seam axis, and that the "+" vertex moves vertically along its local seam surface. Right: view along the seam axis. It proves that also the "o" vertex obeys its local constraint.

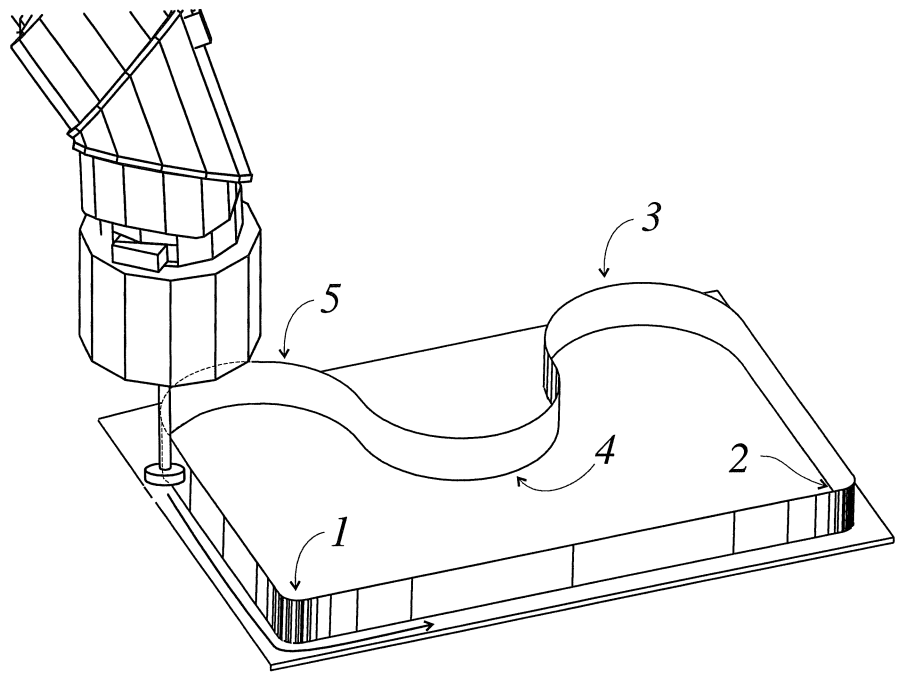


FIGURE 7.13 : *2D contour following: CAD model of contour.*

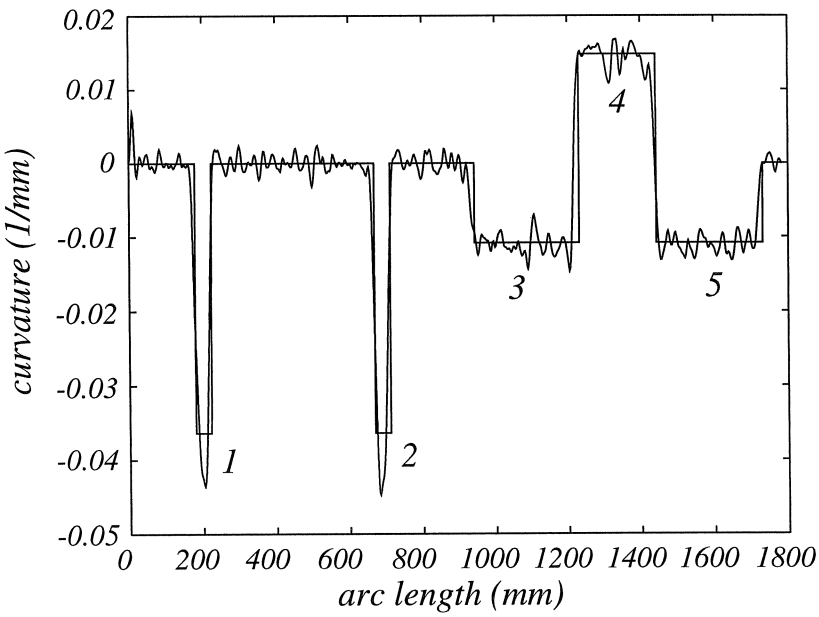


FIGURE 7.14 : *2D contour following: estimated curvature.* The curvature is plotted as a function of the arc length travelled along the contour. The straight lines are the curvature as calculated from a CAD model of the contour. The numbers refer to the corresponding arcs in Fig. 7.13. (Figure courtesy of S. Demey.)

7.5 2D contour following

A 2D contour as in FIG. 7.13 is followed with a *MO* that is a small wheel with radius $r = 12.5mm$. Hence, the *MO* can be modelled as a perfect second order surface. However, the experiment used the simpler vertex approximation, i.e., the limit case for $r = 0$. As proven in Section 6.7, Eq. (6.28), this makes no difference from the first order uncertainty identification point of view. The general-purpose compliant element of FIG. 7.1 —suited for tasks involving forces in all possible directions— is replaced by a brass pipe, which is much stiffer but nevertheless allows a stable force control in the direction normal to the 2D contour. The higher stiffness reduces the disturbing influence of non-measured deformations of the tool compliance.

FIGURE 7.14 shows the on-line estimated curvature, Eq. (6.29), plotted on top of the curvature derived from an analytical CAD model of the contour. The CAD contour is the real contour plus an offset equal to the wheel's radius. It consists of straight lines and circular arcs, hence the curvature is piecewise constant.

A second experiment tries to feedback the estimated curvature to the motion specification: a smaller radius of curvature gives rise to a lower tangential velocity, and vice versa.

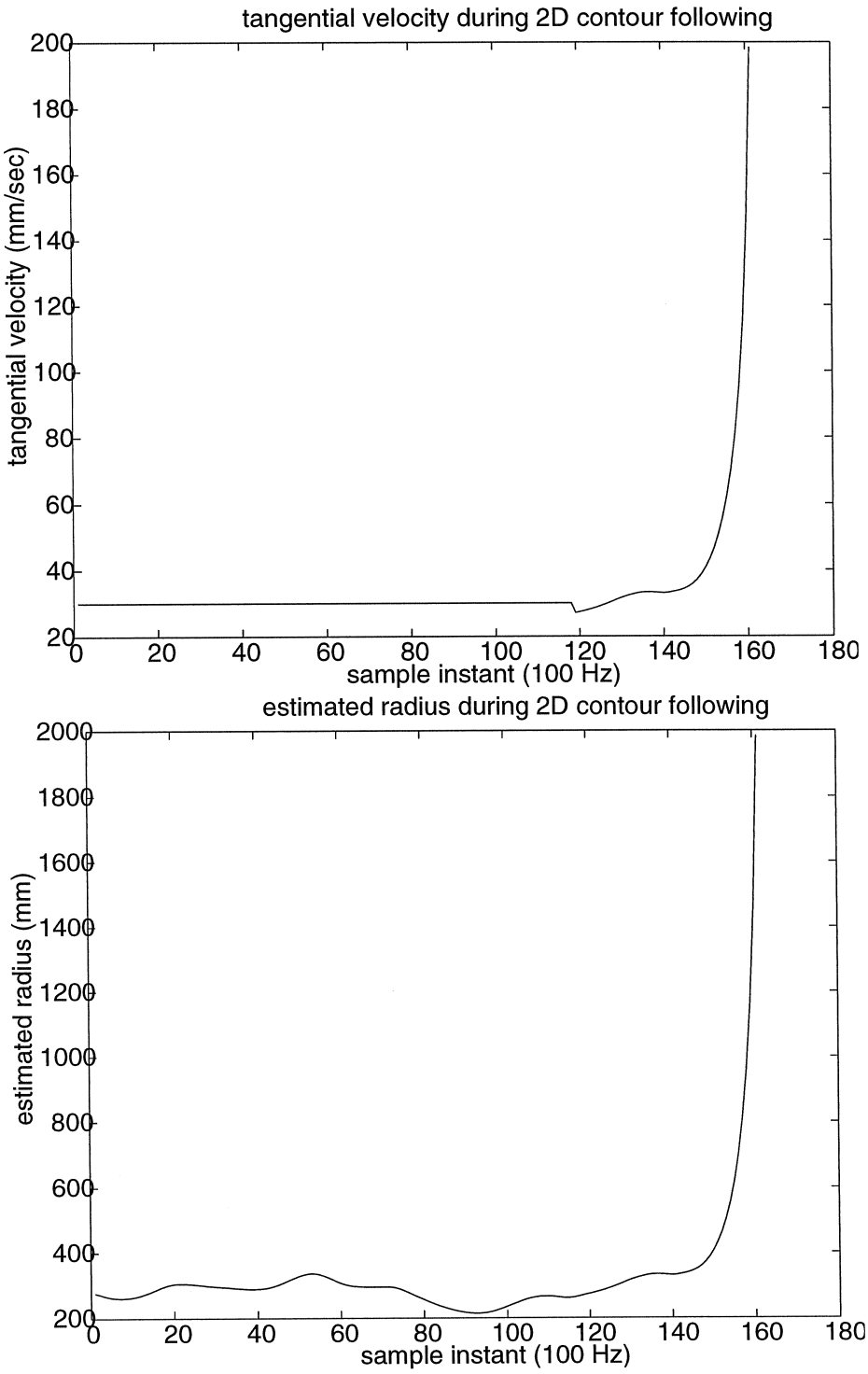


FIGURE 7.15 : *2D contour following: instability in feedback of estimated curvature.* The plots show the specified tangential velocity and the estimated radius. The feedback from estimated radius to tangential velocity is applied starting from about sample period 120.

FIGURE 7.15 shows that this feedback introduces instabilities: the depicted results show the estimated curvature of a cylinder with real radius $r = 299mm$; the curvature is estimated rather accurately, but at a given moment —which differs from execution to execution— the estimate diverges, and the task fails because the specified velocity grows too large and contact is lost. The compliance is presumably the cause of this instability, because:

1. The curvature is derived from the motion of the robot end effector, and not from the motion of the *MO*.

2. These two motions differ by the deformation of the brass element.

3. This deformation is controlled by the force loop.

4. At a certain moment, and due to the interplay of the tracking and control dynamics, the force controller lags behind a bit too much, i.e., the compliant brass element relaxes a bit too far.

5. This implies that the robot end effector temporarily overestimates the radius of the cylinder, which results in an increased tangential velocity due to the proportional feedback between estimated radius and desired tangential velocity.

6. This in turn tends to make the force error even larger, and the instability is excited.

A third experiment works with the second order identification equation (6.37): a cylindrical object with radius $r = 299mm$ is followed with a motion specification corresponding to an erroneous radius $r + \Delta r$. The force controller can cope with the resulting feedforward error, and as a result, the *MO* gets an acceleration a_z with respect to the modelled circular motion. FIGURE 7.16 shows both the nominal and the measured trajectories; FIG. 7.17 shows the relative drift between them. The acceleration of this drift is estimated, after fitting a second order polynomial through the curve. Then, this estimated acceleration a_z is used in Eq. (6.37), repeated here for convenience:

$$\frac{\Delta r}{r} = \frac{a_z}{v_x \omega_y}.$$

The other parameters in this equation are the motion parameters of the nominal trajectory. TABLE 7.1 gives estimated Δr values for different nominal radii r.

7.6 Redundancy resolution

This Section is not a direct application of any material presented in the previous Chapters. However, it shows that using the same kinematic modelling approach, the virtual contact manipulator idea is easily extended to cope with "virtual contacts," i.e., *artificial* constraints on the *MO*'s motion freedom that are not caused by a physical contact

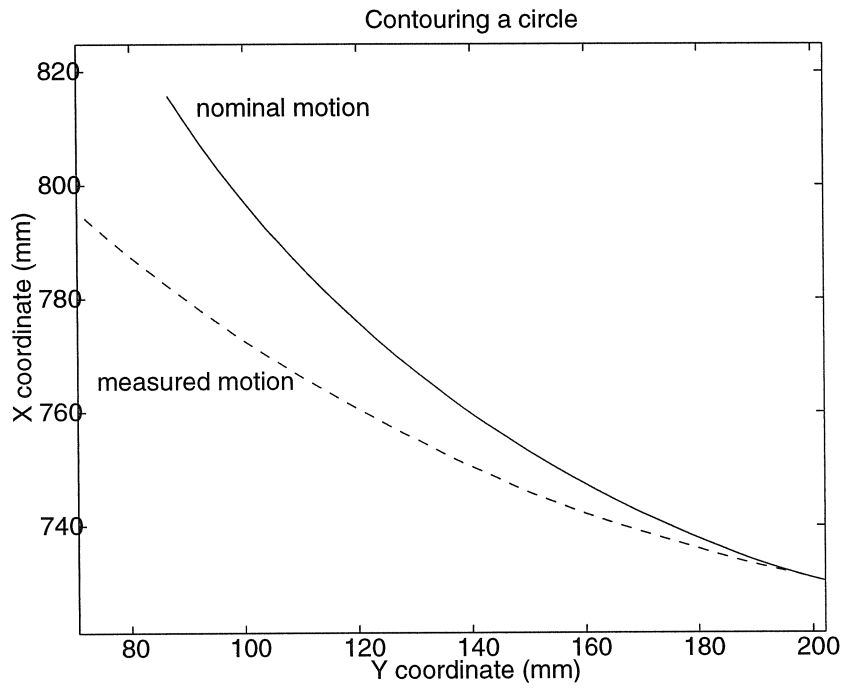


FIGURE 7.16 : *2D contour following: nominal and measured trajectories.* A cylindrical object is followed with a nominal motion specification that uses an incorrect radius, $r = 250mm$, instead of the real radius $r = 299mm$. The plot shows both trajectories over a length of $150mm$. The tangential velocity in both tasks was $29mm/sec$.

nominal radius (mm)	estimated Δr (mm)	total radius (exact: $299mm$)
290	8	298
250	50	300
200	110	310

TABLE 7.1 : *2D contour following of cylindrical object: estimated errors in radius.* The errors Δr are estimated by Eq. (6.37).

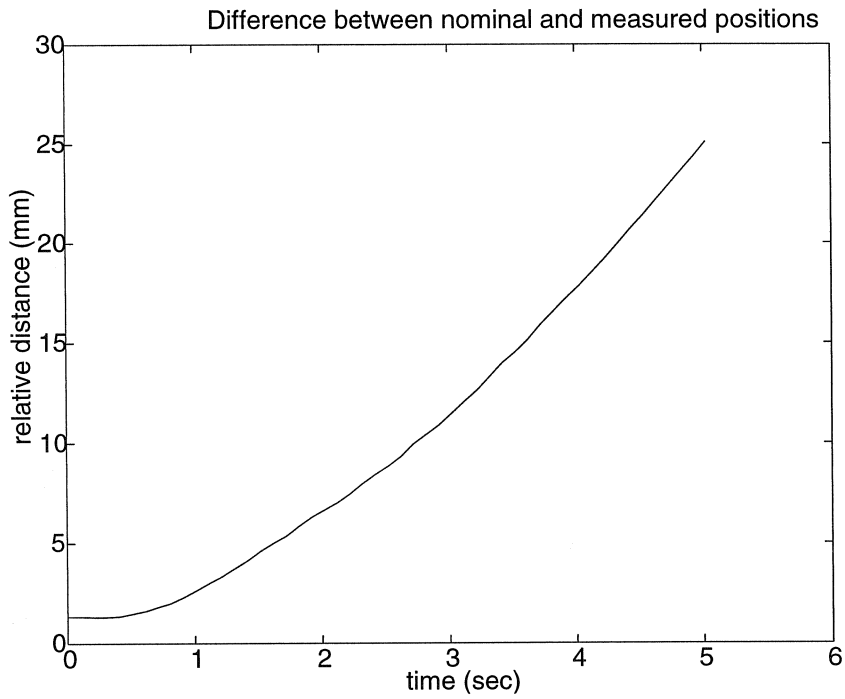


FIGURE 7.17 : *2D contour following: relative drift.* This drift concerns the same trajectories as in FIG. 7.16.

but by performance constraints imposed by the user. Only one example is treated here: redundancy resolution to minimize joint motions.

During the 2D contour following experiment of the previous Section, the orientation of the tool is, in general, not important. Hence, it constitutes a redundant degree-of-freedom. This redundancy can be used to optimize a secondary task, e.g., keeping the joint motions of the manipulator as small as possible. Therefore, the slip joint is connected by a virtual prismatic link and a revolute joint to an "anchor point" somewhere on the robot, FIG. 7.18. Hence, the tool's orientation is now determined by the virtual prismatic joint. The tangential velocity v_x is the only input to the motion specification. It determines the virtual joints' velocity: 1) $\dot{q}_r = v_x / (p\cos(q))$, with p the length of the prismatic link, and q the angle between the direction of the prismatic link and the contact normal; 2) $\dot{p} = -\cos(q)v_x$. The slip velocity set-point is $\dot{q} = \dot{q}_r + \omega_{tr}$, with ω_{tr} the angular velocity about the Y axis of the contact frame, generated by the tracking controller. FIGURE 7.19 shows the measured position of one of the robot joints with and without application of the redundancy resolution procedure. By placing the "anchor point" of the prismatic link somewhere else in space, the example above is easily adapted to use this redundancy resolution for *collision avoidance* purposes: the virtual kinematic chain keeps the tool away from certain orientations that could give rise to collisions, for example, when performing an operation on a concave object. Of course, for the round wheel used in the experiment this collision avoidance procedure would have no effect...

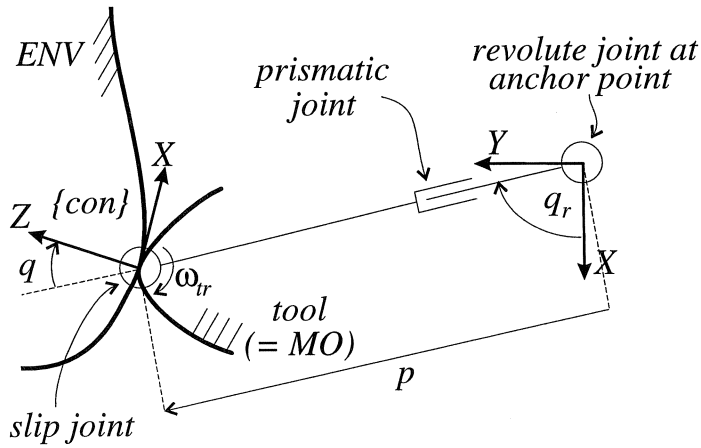


FIGURE 7.18 : *Kinematic model for redundancy resolution.* The slip motion of the tool is determined by the first joint of the robot.

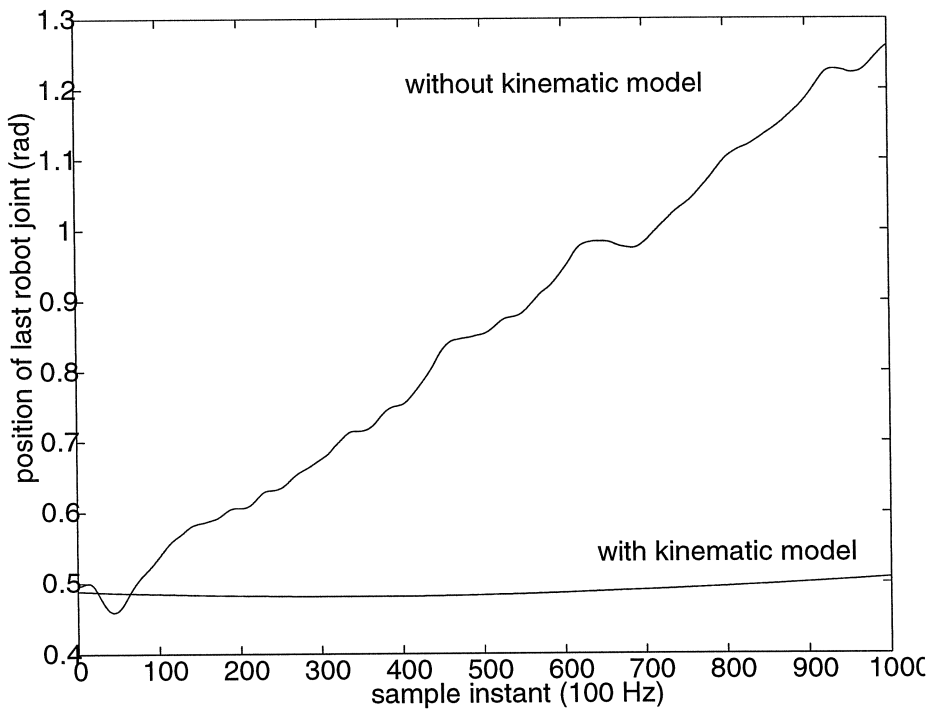


FIGURE 7.19 : *Redundancy resolution: measured joint angle.* The plot shows the measured position of the last robot joint with and without the kinematic redundancy resolution model.

7.7 Suggested potential applications

As already shown in the previous Section, the kinematic approach is easily extended to serve other purposes than only compliant motion. Two more possible application examples are given here: robot calibration and simulation of constrained motion. However, neither of them has been implemented yet.

7.7.1 Robot calibration

2D contour following is an example of the identification of a time varying uncertainty; robot calibration illustrates the identification of time invariant uncertainties. See Bernhardt & Albright (1993) for a recent overview of the state-of-the-art in this domain. The kinematic approach of this thesis, however, generates a possible alternative: the nominal kinematic model of the robot is extended with uncertainty joints, which model the inaccuracies in the knowledge about the robot's geometric structure; for example, a prismatic joint along a link represents inaccurate knowledge of the link length, or two revolute joints orthogonal to every real joint represent orientation errors. For identification purposes, the end effector of the robot is connected to a mechanism in the environment with exactly known kinematics, e.g., a one-dimensional slider rigidly fixed to the environment. The connection between the mechanism and the end effector is compliant, and it is assumed that its deformation can be accurately measured. The robot is now programmed to move the mechanism. Since the robot's nominal kinematics are inaccurate, the resulting motion does not coincide with the mechanism's motion freedom, and the compliant element deforms. Measuring these deformations —which are possibly used to adapt the motion specification for the robot by means of a "force" controller— yields the relative motion of the slider and the robot end effector. And hence the absolute end effector twist $\mathbf{t}_m$ as measured by the robot. On the other hand, the real end effector motion must be compatible with the twist space of the mechanism, or reciprocal to its wrench space. Hence, the first order twist based and wrench based identification approaches from Chapter 6 can be used. If the mechanism imposes a time invariant constraint —such as for example a linear slider— *batch* identification algorithms can be applied.

Compared to the more classical approaches —using theodolites, laser trackers or cameras— this kind of robot calibration offers a real *direct end point measurement.* The classical tools do not really measure at the end point of the robot: they measure the *angle* between an internal reference and the line of sight to the end effector. For the same cost, angle measurements are less accurate than the displacement measurements with which one could measure the deformation of the compliant element. Moreover, position *and* orientation of the robot end effector are directly measured, which is not straightforward with the classical identification devices.

Of course, the presented calibration approach has its own drawbacks too: 1) the calibration slider must be manufactured with sufficiently tight tolerances; 2) various mechanical fixtures are required to position the slider in all relevant parts of the robot's workspace (it is necessary to know the position of these fixtures if one wants to know the absolute

spatial position and orientation of the robot end effector...); and 3) the deformation of the compliant element must be accurately measured. Anyway, such a "self measuring" compliant tool is indispensable for high quality compliant motion tasks.

7.7.2 Simulation

High end graphical workstations begin to get the computing power required to simulate a force controlled robot executing a compliant motion task in real time. For a real world controller, FIG. 1.1, the identification module (i.e., roughly speaking, finding the contact point and contact normal) is one of the toughest parts, and the sensing module (i.e., measure the motion and contact force) one of the simplest. For a simulator, the complexity of these two modules is reversed: the main simulation problem is to develop a good physical contact model, that generates realistic contact forces starting *only* from the geometrical data of the objects. Bruyninckx et al. (1992) present a real time compliant motion simulator with a simple contact model (no friction, no dynamics, no impact) for *convex* and *polyhedral* objects. The kinematic model based approach to compliant motion, described in the previous Chapters, offers possibilities to extend this work to non-polyhedral objects. The following paragraphs present one possible approach (which has not been implemented yet).

In a reaction force simulator, the (ideal) reaction force between two smooth surfaces in contact is calculated as follows:

1. The *MO* moves with a twist t^{MO}, driven by the robot arm. This twist might or might not bring the *MO* in contact with the *ENV*. In a simulation, contact is signaled by the penetration of the *MO*'s geometry with the geometry of the *ENV*.[2]

2. In the neighbourhood of the (imminent) contact between *MO* and *ENV* a kinematic contact model is appended to both objects: each object has two revolute joints (slip for the *MO*, and slide for the *ENV*) that allow the contact point to move over the surfaces of both objects. These two contact points are linked with a prismatic and revolute joint along the line joining these two contact points.

3. The prismatic joint contains a virtual spring, whose stiffness corresponds to the modelled stiffness of the interaction between *MO* and *ENV*.

4. If the objects are not in contact, the spring pulls both "contact" points over the two surfaces closer to each other. (Hence, the same model could be used to generate a trajectory that brings both objects in contact!)

5. If both objects *are* in contact, the spring is compressed. Extremizing the compression energy of the spring brings the modelled contact points to the point of deepest intersection. The ideal contact force then corresponds, in magnitude, to the force in the virtual spring.

[2]Detecting these geometric intersections is a very complex and time-consuming problem, especially for objects with free form surfaces.

7.8 Conclusions

The experiments discussed in this Chapter reveal the following relations between the theoretical developments of the previous chapters on the one hand, and the practical capabilities of a state-of-the-art force controller:

1. The nominal motion specifications generated by the virtual contact manipulators produce very good feedforward information to the force controller.

2. The on-line identification of geometric uncertainties performs well for first order uncertainties and for off-line identification of second order uncertainties; it performs less for on-line identification of second order uncertainties.

Indeed, stability problems arise when the on-line estimates of curvature parameters are fed back to the motion specification. Anyway, higher quality measurements of the real motion of the MO or more robust numerical differentiator algorithms are required if one wants to extract accurate curvature information from the measured accelerations.

The experiments used *instantaneous* identification (and tracking), i.e., the identification is performed *every* sample instant —typically about 150 times per second. Not surprisingly, this turns out to give very "nervous" results, due to the discretisation in the calculations of velocities and accelerations. Non-instantaneous identification approaches, such as fitting a window of measurements to a model of the required order, have not yet been tried out extensively, except for estimating the curvature of a cylindrical object

The kinematic approach to motion constraint modelling and motion specification can be extended to other constraints than contacts, i.e., many user-defined constraints, such as collision avoidance or performance optimization, can be defined by means of artificial, or virtual, "contacts" acting on the MO.

Calibration of the kinematics of a robot, and simulation of constrained motion are other research areas to which the kinematic modelling approach can significantly contribute.

Chapter 8

General conclusions

The previous Chapters presented some theoretical contributions —corroborated by experimental results— towards a more "intelligent" compliant motion system. As already mentioned in the Preface, the attained level of intelligence remains very modest, in this thesis work as well as in all other state-of-the-art research in robotics. It would instead be more realistic to speak about an increased level of "knowledge about the compliant motion problem." In retrospect, that is perhaps the major achievement of this thesis: to have analyzed the concept of "compliant motion" in more detail and with more scrutiny than before; hence, the differences and complementarities between the different components have become more apparent: modelling, motion specification, sensing, identification, monitoring, tracking and set-point control. In particular, this text emphasized the distinctions between model based motion specification ("feedforward") on the one hand, and identification, tracking control and set-point control ("feedback") on the other hand.

The following three Sections give an overview of: 1) the main contributions and conclusions of this thesis work in the area of motion constraint modelling, motion specification and uncertainty identification for compliant motion tasks, Section 8.1; 2) a critical evaluation of the results in the broader framework of sensor controlled robotic manipulation, Section 8.2; and 3) an agenda of future work that is directly related to new problems generated by the presented research.

8.1 Main contributions

This Section lists the parts of this text that are believed to be original contributions to the state-of-the-art in robot compliant motion. The following Section tries to critically examine their impact.

Modelling of motion constraints Starting from the local first and second order geometry of the manipulated object and the environment objects, a kinematic model, the so-called *virtual contact manipulator*, for the general point contact between two smooth and rigid surfaces is constructed. The models for all other contact types are derived as

limit cases. The motion specification and the mathematical representation of the twist and wrench spaces are completely similar to those for normal robotic manipulators.

The virtual contact manipulator model describes the local and reciprocal motion freedom of the *MO* with respect to the *ENV*. It is accurate up to second order, and consistent with the "connection equations" from classical differential geometry. Compared to the latter, it offers a much more user-friendly interface —the *slip, slide* and *rotation* degrees-of-freedom— and is easier to integrate with the other modules of a compliant motion control system.

Uncertainties in the local geometric models are easily incorporated into the kinematic model in the form of uncertainty joints. First order uncertainties represent inaccurate knowledge about the position of the contact point and the direction of the contact normal; they influence the instantaneous motion freedom space spanned by the twist and wrench Jacobians $\boldsymbol{J}$ and $\boldsymbol{G}$. Second order uncertainties represent inaccurate knowledge about the curvature of the contacting surfaces: directions of minimum and maximum curvature, and the corresponding curvature radii. They do not influence the span of $\boldsymbol{J}$ and $\boldsymbol{G}$, but they do change the second order motion freedom models.

For multiple contacts in parallel, each with its own set of uncertainty joints, a systematic procedure is given to reduce the number of uncertainty joints.

The resulting models are intuitive and flexible, yet they are able to deal with multiple, complex and time varying motion constraints between non-polyhedral objects.

Model based motion specification The slip, slide and rotation degrees-of-freedom, as defined by the modelling module, describe all reciprocal degrees of freedom. Hence, specifying a motion of the *MO* by giving velocity set-points for each of these degrees-of-freedom results *by construction* in a motion that is compliant to the (modelled) constraint. Moreover, a simple *first order* specification input, in the form of velocity set-points for the slip, slide and rotation joints, is transformed by the model in a *second order* Cartesian motion specification —i.e., including accelerations— because curvature is explicitly taken into account.

For multiple contacts in parallel the motion is specified by giving velocity set-points to the *driving joints* only. These driving joints form a minimum basis for the twist space of the total constraint, i.e., the intersection of the twist spaces of the individual contacts. This intersection is described by the linear *velocity closure* equations.

Model based motion specification is illustrated by the peg-on-hole experiment.

Identification of geometric uncertainties The *Taylor series expansions* of the Jacobian matrices in terms of the uncertainty parameters are the basis to derive identification equations for an *arbitrary* contact situation and an *arbitrary* set of uncertainty parameters. *Reciprocity* and *consistency* are the physical constraints that reduce the uncertainty search space. The identification equations are called *wrench based* and *twist based*, depending on whether the force or motion measurements are the inputs. Identification equations have been derived for *small* and *time varying* first and second order geometric uncertainties. They allow efficient on-line tracking, as illustrated by the 2D contour following experiment.

New material in manipulator kinematics The *inertial, body-fixed* and *hybrid* representations of twists and wrenches have been described, as well as the transformation formulas between these representations, and between their expressions in different reference frames.

The derivatives of a Jacobian matrix, with respect to time as well as with respect to any joint angle of the manipulator, are calculated symbolically.

8.2 Critical evaluation

This Section discusses the impact of the contributions listed in the previous Section, and tries also to evaluate some indirect results of the presented research work.

Modelling of motion constraints The kinematic approach to describe motion constraint models is only *one possible means* to obtain an integration between modelling, motion specification, sensing and identification. Other approaches could yield the same results, i.e., the same identification equations as derived in Chapter 6, or the same nominal motion specification for the peg-on-hole problem, Appendix D. Anyway, the results can be used independently of the model paradigm that generated them. The major advantages of using kinematic models are: 1) (mechanical) engineers are already quite familiar with kinematics, so the approach is "intuitive"; 2) kinematic models allow to express many aspects of the compliant motion problem in terms of the same modelling paradigm, so *integration* is facilitated; 3) kinematic models are of the same complexity as the problem they describe, i.e., the models are flexible enough to describe all aspects of the compliant motion problem, so it is not necessary to introduce a more complex theoretical tool of which only a small part is finally used —such as is the case for example with non linear control based on Lie algebra and differential geometry; and 4) kinematics is a very mature and active research area, so that (almost) all mathematical tools and techniques are available.

Motion specification The kinematic models, and the motion specification approach based on them, allow and even emphasize a "full" motion specification, i.e., all degrees-of-freedom that are physically possible are treated on the same footing, while the classical Task Frame motion specifications —e.g., seam tracking, contour or surface following— make a (most often implicit) distinction between tracking degrees-of-freedom and the "nominal" degrees-of-freedom. In the tracking degrees-of-freedom, the manipulated object is asked to follow the tracking completely, or else not at all (in which case only the task frame follows the tracking).

Remark that the presented motion specification is limited to be *instantaneous* and *local*. Hence, this research does not contribute solutions to the *planning* problem for compliant motions, i.e., the generation of sequences of compliant motion actions, together with transition conditions and on-line re-planning capabilities to cope with execution errors. Planning involves a more "global" and non-instantaneous motion specification. Nevertheless, both

specification processes are intimately related and complementary.

Identification A first important result of the identification approach based on the kinematic modelling is that all previous "research" on identification —which was in fact based on intuitively derived (but correct) identification equations – is now integrated in a *scientifically* based approach, using physically relevant "first principles," i.e., reciprocity and consistency.

Secondly, it is equally important to realize that full identification of arbitrary 3D contact situations with general and unknown objects is not straightforward if only the measured motion of the manipulated object can be used, together with the measured reaction force. The major problem is to get enough excitation for all uncertainties. For example, it is very difficult to identify all second order uncertainties, especially (the changes in) the principal directions. (These changes are rather *third-order* since they involve the difference between two second order elements!)

Thirdly, not all possible contact situations can be coped with reliably yet. Partly because friction and compliance can cause large disturbances in the force and motion measurements, such that the control system is unable to discern "data" from "noise." This causes especially (major) problems in *monitoring* the contact situation, i.e., it is until now not possible to automatically and reliably detect transitions between contact situations.

Finally, the identification equations derived in Chapter 6 describe *instantaneous* identification. This means that they work with the instantaneous force, velocity and acceleration measurements. Everybody familiar with real world robotics knows that these data can be very noisy, especially for the acceleration measurements. On the other hand, the identification approach remains valid if the identification time scale is taken (much) slower than the control time scale, which allows a more appropriate filtering of the measurements.

New material in manipulator kinematics In retrospect, it is very surprising that the robotics literature contains no publications with an overview of what representations for twists and wrenches are possible, and by what relationships they are connected. Moreover, many publications give no details as to what representation they use. Hence, the material of Chapter 3 is certainly useful far beyond the scope of the research described in this thesis.

Analysis of the compliant motion problem Probably the most important theoretical result of this research is the fact that it *does* give new theoretical results: although force controlled compliant motion is already around for about two decades now, this thesis is among the very few trying to search for scientific foundations of motion constraint identification, while identification is undoubtedly a key aspect of any control system that has to cope with on-line uncertainties.

Another practical (but indirect) result of this research work has been the thorough analysis of the compliant motion problem, resulting in the set of independent but narrowly connected subproblems as depicted in FIG. 1.1, i.e., "compliant motion control" = modelling + specification (local and global; nominal motion and active sensing) + identification (instantaneous and "batch") + monitoring (error and transition detection plus

re-planning) + tracking + motion and force set-point control. In fact, only the control part of the problem has been intensively investigated in the robotics literature.

Invariance of the theory In the previous Chapters of this text, the question of *invariance* of the presented material has never been posed explicitly, but it has been a continuous concern during the development of the theoretical parts. A theory (or rather, its mathematical representation) is called "invariant" (in restricted Euclidean, hence engineering sense) if its physical meaning is conserved under transformations of: 1) the reference frame in which it is expressed, and 2) the physical units of the scalars, vectors and matrices used to describe the theory. As mentioned in Chapter 2, many mistakes due to non-invariance occurred (and still occur!) in the literature on compliant motion and hybrid force control. Thus, it is not unimportant to stress that:

1. The principles of reciprocity and consistency underlying the derivation of identification equations in Chapter 6 do *not* depend on a particular choice of reference frame or physical units.

2. The same remark is valid for the Jacobians used in motion constraint modelling and motion specification.

3. Every equation in the text that makes explicit use of a particular reference frame —e.g., the expressions for the Jacobians with respect to the principal frames of MO and ENV; or all explicit identification equations derived for a set of typical contact situations— can be transformed to whatever other reference frame, by means of the screw transformation matrices introduced in Chapter 3, without changing its physical meaning.

8.3 Future work

The previous Sections highlighted already the most important gaps between the theoretical results on the one hand, and their real world implementation on the other hand: firstly, sensor processing to extract relevant information from the noisy and disturbed raw data, taking into account the information contained in the model, and secondly, planning and control of active sensing motions, superimposed on top of the nominal motion, in order to excite all uncertainties. The following paragraphs sketch outlines of what further research is needed to solve these problems.

Model based sensor processing The sensor information for a force controlled compliant motion system comes from two sources: 1) position and/or velocity information from the robot, and 2) force/torque information from the wrist force sensor.

In a setup with a very stiff robot arm and with the mechanical compliance mainly located between the robot end effector and the contact, the measured position information is disturbed by the deformation of this mechanical compliance. A solution to this problem

could be to locate the major source of compliance in or next to the force sensor, and to put position sensors around it. In this way, the passive deformation of the compliance is measured, and together with the measured robot position this should give much more accurate motion information. This approach is followed in our laboratory, Degezelle & Peeters (1994), but the developed device was not fully operational yet at the time of writing. Preliminary tests however show a much improved position measurement capability. On the other hand, such a flexible force sensor requires an adaptation of the actual robot control software.

To detect a transition between two different contact situations seems simple at first sight, i.e., one just has to look for sudden changes in one or more force or motion components, or in the identified uncertainties. However, looking at real data reveals the real world problems: 1) the data is full of "sudden changes," caused by sensor noise, stick-slip, surface texture, etc., and 2) the transition can only be determined reliably "long" after it actually occurred ("long" with respect to the robot control frequency). One appropriate way to cope with these problems is probably to be found in stochastic estimation, such as (Extended) Kalman filtering. The current and expected contact models then serve as a basis for the design of the filters.

Active sensing If the nominal motion of the manipulated object does not sufficiently excite the uncertainties, the following two solution strategies present themselves:

1. *Spatial* parallelism: use a specially designed end effector, that surrounds the *MO* and contacts the environment with a number of closely spaced "fingers," with each of these fingers having its own passive and compliant degree-of-freedom, approximately normal to the surface. Measuring the normal deformation of each of the fingers, as well as its time derivative, gives similar information as for one single contact on the *MO*, but now at several contact points *at the same time.* Hence, the previously underdetermined identification equations now get multiple inputs for the same uncertainties. A good trade-off must be found between the number of fingers, their relative spacing and the expected curvature of the environment. Of course, such a solution is only possible for small manipulated objects.

   Examples of such multi-fingered probes are hardly reported in the literature; e.g., Stute & Erne (1979). However, tactile arrays are very similar in principle. Reynaerts & Van Brussel (1993) describe successful estimation of the curvature of an object grasped by a robot hand, by means of the tactile array mounted on the finger surface.

2. *Temporal* parallelism: if it is not possible to instantaneously move the contact in all directions required for full identification, one could regularly interrupt the nominally specified trajectory of the manipulated object, in order to schedule "active sensing" actions that move the object also in the directions not excited by the nominal motion. The research problems here are: What active sensing actions are required? When are they to be scheduled? How can one make sure that all active sensing actions work with exactly the same contact situation?

In practice, neither of these solutions has been thoroughly examined yet.

Bibliography

Ahmad, S. and Lee, C. N., 1990, *Shape recovery from robot contour-tracking with force feedback*, Proceedings IEEE International Conference on Robotics and Automation, pp. 447–452.

Ando, S. and Miyake, N., 1981, *Development of a new robot profile following method*, Proceedings of the International Symposium on Industrial Robots, pp. 447–454.

Angeles, J., 1988, *Rational Kinematics*, Springer-Verlag.

Angeles, J., 1993, *Robotic Mechanical Systems*, A Set of Lecture Notes, McGill University, Montreal, Canada.

Angeles, J., 1994, *On twist and wrench generators and annihilators*, In "Computer-Aided Analysis of Rigid and Flexible Mechanical Systems," Edited by Seabra Pereira, M. F. O. and Ambrósio, J. A. C., NATO ASI Series, Series E: Applied Sciences, Vol. 268, Kluwer Academic Publishers, pp. 379–411.

Asada, H. and Hanafusa, H., 1980, *Spatial curve tracing control of industrial robots*, Trans. Soc. Instrum. and Control Eng., Vol. 16, pp. 740–746.

Ball, R. S, 1900, *A Treatise on the Theory of Screws*, Cambridge University Press.

Bennett, D. J. and Hollerbach, J. M., 1989, *Identifying the kinematics of robots and their tasks*, Proceedings IEEE Conference on Robotics and Automation, pp. 580–586.

Bernhardt, R. and Albright, S. L., 1993, *Robot Calibration*, Chapman & Hall.

Bicchi, A., Salisbury, J. K. and Brock, D. L., 1993, *Contact sensing from force measurements*, The International Journal of Robotics Research, Vol. 12, No. 3, pp. 249–262.

Blauer, M. and Bélanger, P. R., 1987, *State and parameter estimation for robotic manipulators using force measurements*, IEEE Transactions on Automatic Control, Vol. AC-32, No. 12, pp. 1055–1066.

Bottema, A. and Roth, B., 1979, *Theoretical Kinematics*, North-Holland, Amsterdam.

Bruyninckx H., De Schutter J, and Allotta B., 1991a, *Model-based constrained motion: a. modelling, specification and control*, Proceedings of ICAR'91, International Conference on Advanced Robotics, pp. 976–981.

Bruyninckx H., De Schutter J, and Allotta B., 1991b, *Model-based constrained motion: b. introducing on-line identification and observation of the motion constraints*, Proceedings of ICAR'91, International Conference on Advanced Robotics, pp. 982–987.

Bruyninckx, H, and De Schutter, J., 1992, *A systematic derivation of on-Line motion constraint identification equations for model-based compliant motions*, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1513–1518.

Bruyninckx, H, De Schutter, J., van de Poel, P, and Witvrouw, W., 1992, *A CAD-based contact force simulator as a learning tool for compliant motions*, International Conference on Intelligent Control, Glasgow, pp. 287–292.

Bruyninckx, H, and De Schutter, J., 1993, *Kinematic models for rigid body interactions for compliant motion tasks in the presence of uncertainties*, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1007-1012.

Bruyninckx, H., De Schutter, J. and Dutré S., 1993, *The "reciprocity" and "consistency" based approaches to uncertainty identification for compliant motions*, Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, pp. 349–354.

Bruyninckx, H, Dutré, S. and De Schutter, J., 1995, *Peg-on-hole : a model based solution to peg and hole alignment*, Proceedings of the IEEE International Conference on Robotics and Automation, accepted for publication.

Cai, C. S. and Roth, B., 1986, *On the planar motion of rigid bodies with point contact*, Mechanism and Machine Theory, Vol. 21, No. 6, pp. 453–466.

Cai, C. S. and Roth, B., 1987, *On the spatial motion of a rigid body with point contact*, IEEE International Conference on Robotics and Automation, pp. 686–695.

Cai, C. S. and Roth, B., 1988, *On the spatial motion of a rigid body with line contact*, Proceedings IEEE International Conference on Robotics and Automation, pp. 1036–1041.

Cai, C., 1990, *Control of a manipulator under multiple contacts between rigid curved surfaces*, Proceedings IEEE International Conference on Robotics and Automation, pp. 1356–1361.

Craig, J. J., 1986, *Introduction to Robotics*, Addison-Wesley.

Degezelle, P. and Peeters, B., 1994, *Design of a compliant force sensor for robotic applications, with optical measurement of the deformation* (in Dutch), Master's Thesis 94EP32, Dept. Mechanical Engineering, Katholieke Universiteit Leuven, Belgium.

Demey, S. and De Schutter, J., 1994, *Enhancing surface following with invariant differential part models*, Proceedings IEEE International Conference on Robotics and Automation, pp. 668–673.

Denavit, J. and Hartenberg, R. S., 1955, *Kinematic notation for lower-pair mechanisms based on matrices*, Transactions of the ASME, Journal of Applied Mechanics, Vol. 23, pp. 215–221.

Desai, R. S. and Volz, R. A., 1989, *Identification and verification of termination conditions in fine motion in presence of sensor errors and geometric uncertainties*, Proceedings IEEE International Conference on Robotics and Automation, pp. 800–807.

De Schutter, J., 1986, *Compliant Robot Motion: Task Formulation and Control*, PhD. Thesis, Katholieke Universiteit Leuven, Department of Mechanical Engineering, Belgium.

De Schutter, J., 1988, *Improved force control laws for advanced tracking applications*, Proceedings IEEE International Conference on Robotics and Automation, pp. 1497–1502.

De Schutter, J., Bruyninckx, H. and Dutré, S., 1993, *Application of computer aided kinematics to modelling of contacts in robotic manipulation*, Proceedings Nato Advanced Study Institute on Computer Aided Analysis of Rigid and Flexible Mechanical Systems, pp. 413-444.

De Schutter, J. and Van Brussel, H., 1988a, *Compliant robot motion I. A formalism for specifying compliant motion tasks*, The International Journal of Robotics Research, Vol. 7, No. 4, pp. 3–17.

De Schutter, J. and Van Brussel, H., 1988b, *Compliant robot motion II. A control approach based on external control loops*, The International Journal of Robotics Research, Vol. 7, No. 4, pp. 18–33.

Driels, M. R., 1986, *A symbolic matrix manipulation package for the kinematic analysis of robot manipulators*, Computers in Mechanical Engineering, Vol. 5, pp. 38–46.

Duffy, J., 1990, *The Fallacy of modern hybrid control theory that is based on "orthogonal complements" of twists and wrench spaces*, Journal of Robotic Systems, Vol. 7, No. 2, pp. 139–144.

Featherstone, R., 1983, *Position and velocity transformations between robot end-effector coordinates and joint angles*, The International Journal of Robotics Research, Vol. 2, No. 2, pp. 35–45.

Gerelle, E. G. R., 1978, *Force feedback control*, Proceedings of the International Symposium on Industrial Robots, pp. 194–205.

Griffis, M. and Duffy, J., 1991, *Kinestatic control: a novel theory for simultaneously regulating force and displacement*, Transactions of the ASME, Journal of Mechanical Design, Vol. 113, pp. 508–515.

Hanafusa, H., Yoshikawa, T., Nakamura, Y. and Takeda, M., 1981, *Contouring control of an articulated robot arm with manipulation variable feedback*, Proceedings of the International Symposium on Industrial Robots, Tokyo, Japan, pp. 429–436.

Haug, E. J., 1989, *Computer Aided Kinematics and Dynamics of Mechanical Systems*, Allyn and Bacon.

Hirai, S. and Asada, H., 1993, *Kinematics and statics of manipulation using the theory of polyhedral convex cones*, The International Journal of Robotics Research, Vol. 12, No. 5, pp. 434–447.

Hirai, S. and Iwata, K., 1992, *Recognition of contact state based on geometric model*, Proceedings International Conference on Robotics and Automation, pp. 1507–1512.

Hogan, N., 1985, *Impedance control: an approach to manipulation, Part I, Part II, Part III*, Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, pp. 1–24.

Hunt, K. H., 1990, *Kinematic Geometry of Mechanisms*, Oxford Science Publications.

Karger, A. and Novák, J., 1985, *Space Kinematics and Lie Groups*, Gordon and Breach New York (N.Y.).

Kazanzides, P., Bradley, N. S. and Wolovich, W. A., 1989, *Dual-drive force/velocity control: implementation and experimental results*, Proceedings IEEE International Conference on Robotics and Automation, pp. 92–97.

Kazerooni, H., Houpt, P. K. and Sheridan, T. B., 1986, *Robust compliant motion for manipulators*, IEEE Journal of Robotics and Automation, RA-2, pp. 83–105.

Kerr, J. and Roth, B., 1986, *Analysis of multifingered hands*, The International Journal of Robotics Research, Vol. 4, No. 4, pp. 3–17. Robotics Research, Vol. 4, No. 1, pp. 3–12.

Kobayashi, H., 1985, *Control and geometrical considerations for an articulated robot hand*, The International Journal of Robotics Research, Vol. 4, No. 1, pp. 3–12.

Laugier, Ch., 1989, *Planning fine motion strategies by reasoning in the contact space*, Proceedings IEEE International Conference on Robotics and Automation, pp. 653–661.

Lee, S. and Asada, H., 1994, *Assembly of parts with irregular surfaces using active force sensing*, Proceedings IEEE International Conference on Robotics and Automation, pp. 2639–2644.

Lipkin, H. and Duffy, J., 1988, *Hybrid twist and wrench control for a robotic manipulator*, Transactions of the ASME, Journal of Mechanisms, Transmissions and Automation in Design, Vol. 110, pp. 138–144.

Lončarić, J., 1987, *Normal forms of stiffness and compliance matrices*, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 6, pp. 567–572.

Lu, J., Duffie, N. A. and Bollinger, J. G., 1982, *Two dimensional tracing and measurement using touch trigger probes*, Annals of the CIRP, Vol. 31/1, pp. 415–419.

Mason, M. T., 1981, *Compliance and force control for computer controlled manipulators*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11, No. 6, pp. 418–432.

McCarragher, B. J., 1993, *Task-level adaptation using a discrete event controller for robotic assembly*, Proceedings of the 1993 International Conference on Intelligent Robots and Systems, Yokohama, Japan, pp. 2281–2285.

McCarragher, B. J. and Asada, H., 1993, *Qualitative template matching using dynamic process models for state transition recognition of robotic assembly*, Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 115, pp. 261–269.

Mimura, N. and Funahashi, Y., 1994, *Parameter identification of contact conditions by active force sensing*, Proceedings IEEE International Conference on Robotics and Automation, pp. 2645–2650.

Mirman, C. R. and Gupta, K. C., 1993, *Identification of position-independent robot parameter errors using special Jacobian matrices*, The International Journal of Robotics Research, Vol. 12, No. 3, pp. 288–298.

Montana, D.J., 1988, *The kinematics of contact and grasp*, International Journal of Robotics Research, Vol. 7, No. 3, pp. 17–32.

Ohwovoriole, M. S. and Roth, B., 1981, *An extension of screw theory*, Transactions of the ASME, Journal of Mechanical Design, Vol. 103, No. 4, pp. 725–735.

O'Neill, B., 1966, *Elementary Differential Geometry*, Academic Press.

Orin, D. E. and Schrader, W. W., 1984, *Efficient computation of the Jacobian for robot manipulators*, The International Journal of Robotics Research, Vol. 3, No. 4, pp. 66–75.

Paljug, E., Yun, X. and Kumar, V., 1994, *Control of rolling contacts in multi-arm manipulation*, IEEE Transactions on Robotics and Automation, Vol. 10, No. 4, pp. 441–452.

Park, F., 1994, *Distance metrics on the rigid-body motions with applications to mechanism design*, Internal Report, Department of Mechanical and Aerospace Engineering, University of California at Irvine.

Park, F., and Bobrow, J., 1994, *A recursive algorithm for robot dynamics using Lie groups*, Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, pp. 1535–1540.

Patarinski, S. P. and Botev, R. G., 1993, *Robot force control: a review*, Mechatronics, Vol. 3, No. 4, pp. 377-398.

Paul, R. P., 1986, *Robot Manipulators: Mathematics, Programming and Control*, MIT Press, Cambridge, Mass., USA.

Raibert, M. H. and Craig, J. J., 1981, *Hybrid position/force control of manipulators*, Transactions of the ASME, Journal of Dynamic Systems, Measurements and Control, Vol. 103, pp. 126–133.

Reynaerts, D. and Van Brussel, H., 1993a, *Tactile sensing data interpretation for object manipulation*, Sensors and Actuators A, Vol. 37-38, pp. 268–273.

Reynaerts, D. and Van Brussel, H. 1993b, *Two-fingered full envelope dexterous manipulation*, Proceedings IEEE International Conference on Robotics and Automation, Atlanta, pp. 436–441.

Rimon, E. and Burdick, J., 1993, *Towards planning with force constraints: on the mobility of bodies in contact*, Proceedings IEEE International Conference on Robotics and Automation, Atlanta, pp. 994–1000.

Rimon, E. and Burdick, J., 1994, *Mobility of bodies in contact, I-II*, Proceedings IEEE International Conference on Robotics and Automation, San Diego, pp. 2329–2341.

Sarkar, N., Yun, X. and Kumar, V., 1993, *Dynamic control of 3-D rolling contacts in two-arm manipulation*, Proceedings IEEE International Conference on Robotics and Automation, Vol. 3, pp. 978–983.

Simons, J., Van Brussel, H., De Schutter, J. and Verhaert, J., *A self-learning automaton with variable resolution for high precision assembly by industrial robots*, IEEE Transactions on Automatic Control, AC-27, No. 5, pp. 1109–1113.

Strang, G., 1988, *Linear Algebra and its Applications, 3rd ed.*, Harcourt Brace Jovanovich, San Diego.

Stute, G. and Erne, H., 1979, *The control design of an industrial robot with advanced tactile sensitivity*, Proceedings of the International Symposium on Industrial Robots, pp. 519–527.

Van Brussel, H. and Simons, J., 1979, *The adaptable compliance concept and its use for automatic assembly by active force feedback accomodations*, Proceedings of the International Symposium on Industrial Robots, pp. 167–181.

Van de Poel, P., Witvrouw, W., Bruyninckx, H. and De Schutter, J., 1993, *An environment for developing and optimizing compliant robot motion tasks*, Proceedings '93 ICAR (International Conference on Advanced Robotics), November 1-2, Tokyo, Japan, pp. 713–718.

Wampler, C., 1984, *Multiprocessor control of a telemanipulator with optical proximity sensors*, The International Journal of Robotics Research, Vol. 3, No. 1, pp. 40–50.

West, H. and Asada, H., 1985, *A method for the design of hybrid position/force controllers for manipulation constrained by contact with the environment*, Proceedings IEEE International Conference on Robotics and Automation, pp. 251–259.

Whitney, D. E., 1969, *Resolved motion rate control of manipulators and human prostheses*, IEEE Transactions on Man-Machine Systems, Vol. 10, No. 2, pp. 47–52.

Whitney, D. E., 1972, *The mathematics of coordinated control of prosthetic arms and manipulators*, Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 94, pp. 303–309.

Whitney, D. E., 1987, *Historic perspective and state of the art in robot force control*, The International Journal of Robotics Research, Vol. 6, No. 1, pp. 3–14.

Xiao, J. and Volz, R.A., 1988, *Design and motion constraints of part-mating planning in the presence of uncertainties*, Proceedings IEEE International Conference on Robotics and Automation, pp. 1260–1268.

Xiao, J., 1992, *Replanning with compliant rotation in the presence of uncertainties*, International Symposium on Intelligent Control, Glasgow pp. 102–108.

Yoshikawa T. and Sudou A., 1993, *Dynamic hybrid position/force control of robot manipulators —on-line estimation of unknown constraint*, IEEE Transactions on Robotics and Automation, Vol. 9, No. 2, pp. 220–226.

Appendix A

Derivatives for the surface-surface contact

This Appendix calculates the twist and wrench Jacobians for the virtual contact manipulator of the *surface-surface* contact, as well as their derivatives with respect to their joint angles, in all three representation types discussed in Chapter:3.

A.1 Twist and wrench Jacobians

This Appendix chooses the inertial frame $\{bs\}$ and the end effector frame $\{ee\}$ to coincide with the principal frame $\{pri\}$. This means that the twist Jacobians in all three representations, ${}_{bs}\boldsymbol{J}^{bs}$, ${}_{ee}\boldsymbol{J}^{ee}$ and ${}_{bs}\boldsymbol{J}^{ee}$, are all equal, and given by Eqs.:(4.3)-(4.5)::

$$ {}_{pri}\boldsymbol{J} = \left[{}^{pri'}_{:pri}\boldsymbol{S}\ {}_{pri'}\boldsymbol{J}'^{+}\ \ {}^{pri'}_{:pri}\boldsymbol{S}\ {}_{pri'}\boldsymbol{J}'^{-}\ \ {}_{pri}\boldsymbol{J}^{r}\ \ {}_{pri}\boldsymbol{J}^{+}\ \ {}_{pri}\boldsymbol{J}^{-} \right], \tag{A.1} $$

$$ \text{with}\ \ {}_{pri'}\boldsymbol{J}'^{+} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -r'^{+} \\ 0 \\ 0 \end{bmatrix}, \quad {}_{pri'}\boldsymbol{J}'^{-} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ r'^{-} \\ 0 \end{bmatrix}, $$

$$ {}_{pri}\boldsymbol{J}^{r} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}_{pri}\boldsymbol{J}^{+} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -r^{+} \\ 0 \\ 0 \end{bmatrix}, \quad {}_{pri}\boldsymbol{J}^{-} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ r^{-} \\ 0 \end{bmatrix}. $$

The wrench Jacobian is equal to the rotation part of the twist Jacobian:: ${}_{pri}\boldsymbol{G} = {}_{pri}\boldsymbol{J}^{r}$. ${}^{pri'}_{:pri}\boldsymbol{S}$ is a simple transformation:: $\{pri\}$ and $\{pri'\}$ have coinciding origins, opposite Z directions, and their X axes are rotated over an angle α about the Z axis. Hence::

$$ {}^{pri'}_{:pri}\boldsymbol{S} = \begin{bmatrix} c & s & 0 & 0 & 0 & 0 \\ s & -c & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & s & 0 \\ 0 & 0 & 0 & s & -c & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \tag{A.2} $$

with $c = \cos(\alpha)$ and $s = \sin(\alpha)$. Then::

$$ {}_{pri}\boldsymbol{J}'^{+} = \begin{bmatrix} s \\ -c \\ 0 \\ -r'^{+}c \\ -r'^{+}s \\ 0 \end{bmatrix}, \quad {}_{pri}\boldsymbol{J}'^{-} = \begin{bmatrix} c \\ s \\ 0 \\ r'^{-}s \\ -r'^{-}c \\ 0 \end{bmatrix}. \tag{A.3} $$

A.2 Inertial representation

Equation:(3.36) is used to calculate the partial derivatives. All Jacobians in this Section are in the inertial representation, but the "bs" subscripts and superscripts are omitted for brevity. According to TABLE:3.2, only the partial derivatives of $\boldsymbol{J}^{r}, \boldsymbol{J}^{+}$ and $\boldsymbol{J}^{-}$ with respect to q'^{+} and q'^{-} differ from zero, as well as the partial derivative of $\boldsymbol{J}^{+}$ and $\boldsymbol{J}^{-}$ with respect to q^{r}. One easily calculates the $\boldsymbol{S}_{\Delta}\left(\boldsymbol{J}^{j}\right), j \in \{'+, '-, r\}$ needed in Eq.:(3.36)::

$$ \boldsymbol{S}_{\Delta}\left(\boldsymbol{J}'^{+}\right) \overset{(3.31)}{=} \begin{bmatrix} 0 & 0 & -c & 0 & 0 & 0 \\ 0 & 0 & -s & 0 & 0 & 0 \\ c & s & 0 & 0 & 0 & 0 \\ 0 & 0 & -r'^{+}s & 0 & 0 & -c \\ 0 & 0 & r'^{+}c & 0 & 0 & -s \\ r'^{+}s & -r'^{+}c & 0 & c & s & 0 \end{bmatrix}, \tag{A.4} $$

$$ \boldsymbol{S}_{\Delta}\left(\boldsymbol{J}'^{-}\right) = \begin{bmatrix} 0 & 0 & s & 0 & 0 & 0 \\ 0 & 0 & -c & 0 & 0 & 0 \\ -s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & -r'^{-}c & 0 & 0 & s \\ 0 & 0 & -r'^{-}s & 0 & 0 & -c \\ r'^{-}c & r'^{-}s & 0 & -s & c & 0 \end{bmatrix}, \tag{A.5} $$

$$ \boldsymbol{S}_{\Delta}\left(\boldsymbol{J}^{r}\right) = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.6} $$

The resulting derivatives are::

$$ \frac{\partial \boldsymbol{J}^{r}}{\partial q'^{+}} = \frac{\partial \boldsymbol{G}}{\partial q'^{+}} = \begin{bmatrix} -c \\ -s \\ 0 \\ -r'^{+}s \\ r'^{+}c \\ 0 \end{bmatrix}, \frac{\partial \boldsymbol{J}^{r}}{\partial q'^{-}} = \frac{\partial \boldsymbol{G}}{\partial q'^{-}} = \begin{bmatrix} s \\ -c \\ 0 \\ -r'^{-}c \\ -r'^{-}s \\ 0 \end{bmatrix}, \tag{A.7} $$

$$\frac{\partial \boldsymbol{J}^{+}}{\partial q'^{+}} = \begin{bmatrix} 0 \\ 0 \\ -s \\ 0 \\ 0 \\ -c\left(r^{+}+r'^{+}\right) \end{bmatrix}, \frac{\partial \boldsymbol{J}^{+}}{\partial q'^{-}} = \begin{bmatrix} 0 \\ 0 \\ c \\ 0 \\ 0 \\ s\left(r^{+}+r'^{-}\right) \end{bmatrix}, \frac{\partial \boldsymbol{J}^{+}}{\partial q^{r}} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -r^{+} \\ 0 \end{bmatrix}, \tag{A.8}$$

$$\frac{\partial \boldsymbol{J}^{-}}{\partial q'^{+}} = \begin{bmatrix} 0 \\ 0 \\ c \\ 0 \\ 0 \\ s\left(r^{-}+r'^{+}\right) \end{bmatrix}, \frac{\partial \boldsymbol{J}^{-}}{\partial q'^{-}} = \begin{bmatrix} 0 \\ 0 \\ -s \\ 0 \\ 0 \\ c\left(r^{-}+r'^{-}\right) \end{bmatrix}, \frac{\partial \boldsymbol{J}^{-}}{\partial q^{r}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -r^{-} \\ 0 \\ 0 \end{bmatrix}. \tag{A.9}$$

A.3 Body-fixed representation

Equation:(3.37) is used to calculate the partial derivatives. All Jacobians in this Section are in the body-fixed representation, but the "ee" subscripts and superscripts are omitted. According to TABLE:3.2, only the partial derivatives of $\boldsymbol{J}'^{+}$ and $\boldsymbol{J}'^{-}$ with respect to q^{+}, q^{-} and q^{r} differ from zero. Hence, Eq.:(3.37) needs the following $\boldsymbol{S}_{\Delta}\left(\boldsymbol{J}^{j}\right), j \in \{r,+,-\}$::

$$\boldsymbol{S}_{\Delta}\left(\boldsymbol{J}^{r}\right) = \text{Eq.} : (\text{A.6}),$$

$$\boldsymbol{S}_{\Delta}\left(\boldsymbol{J}^{+}\right) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & r^{+} & 0 & 0 & 0 \\ 0 & -r^{+} & 0 & -1 & 0 & 0 \end{bmatrix}, \tag{A.10}$$

$$\boldsymbol{S}_{\Delta}\left(\boldsymbol{J}^{-}\right) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & r^{-} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ -r^{-} & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{A.11}$$

Then, the non-zero derivatives are::

$$\frac{\partial \boldsymbol{J}'^{+}}{\partial q^{r}} = \begin{bmatrix} -c \\ -s \\ 0 \\ -r'^{+}s \\ r'^{+}c \\ 0 \end{bmatrix}, \frac{\partial \boldsymbol{J}'^{+}}{\partial q^{+}} = \begin{bmatrix} 0 \\ 0 \\ s \\ 0 \\ 0 \\ -c\left(r^{+}+r'^{+}\right) \end{bmatrix}, \frac{\partial \boldsymbol{J}'^{+}}{\partial q^{-}} = \begin{bmatrix} 0 \\ 0 \\ -c \\ 0 \\ 0 \\ s\left(r^{-}+r'^{+}\right) \end{bmatrix}, \tag{A.12}$$

$$\frac{\partial \boldsymbol{J}^r}{\partial q^+} = \frac{\partial \boldsymbol{G}}{\partial q^+} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -r^+ \\ 0 \end{bmatrix}, \frac{\partial \boldsymbol{J}^r}{\partial q^-} = \frac{\partial \boldsymbol{G}}{\partial q^-} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -r^- \\ 0 \\ 0 \end{bmatrix}, \tag{A.13}$$

$$\frac{\partial \boldsymbol{J}'^-}{\partial q^r} = \begin{bmatrix} s \\ -c \\ 0 \\ -r'^- c \\ -r'^- s \\ 0 \end{bmatrix}, \frac{\partial \boldsymbol{J}'^-}{\partial q^+} = \begin{bmatrix} 0 \\ 0 \\ c \\ 0 \\ 0 \\ s\left(r^+ + r'^-\right) \end{bmatrix}, \frac{\partial \boldsymbol{J}'^-}{\partial q^-} = \begin{bmatrix} 0 \\ 0 \\ -s \\ 0 \\ 0 \\ c\left(r^- + r'^-\right) \end{bmatrix}. \tag{A.14}$$

A.4 Hybrid representation

Equations:(3.39)-(3.40) are used to calculate the partial derivatives. All Jacobians in this Section are in the hybrid representation, but the "bs" subscripts and the "ee" superscripts are omitted. TABLE:3.2 shows that most partial derivatives now differ from zero. Since $\boldsymbol{S}_\Delta = \boldsymbol{P}_\Delta + \boldsymbol{M}_\Delta$, Eqs.:(A.4)-(A.5) and (A.6)-(A.11) give the $\boldsymbol{P}_\Delta\left(\boldsymbol{J}^j\right)$ and $\boldsymbol{M}_\Delta\left(\boldsymbol{J}^j\right)$ needed to calculate the derivatives::

$$\frac{\partial \boldsymbol{J}^+}{\partial q^r} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -r^+ \\ 0 \end{bmatrix}, \frac{\partial \boldsymbol{J}^+}{\partial q^+} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^+ \end{bmatrix}, \frac{\partial \boldsymbol{J}^+}{\partial q'^+} = \begin{bmatrix} 0 \\ 0 \\ s \\ 0 \\ 0 \\ -r^+ c \end{bmatrix}, \frac{\partial \boldsymbol{J}^+}{\partial q'^-} = \begin{bmatrix} 0 \\ 0 \\ c \\ 0 \\ 0 \\ r^+ s \end{bmatrix}, \tag{A.15}$$

$$\frac{\partial \boldsymbol{J}^-}{\partial q^r} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -r^- \\ 0 \\ 0 \end{bmatrix}, \frac{\partial \boldsymbol{J}^-}{\partial q^-} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^- \end{bmatrix}, \frac{\partial \boldsymbol{J}^-}{\partial q'^+} = \begin{bmatrix} 0 \\ 0 \\ c \\ 0 \\ 0 \\ r^- s \end{bmatrix}, \frac{\partial \boldsymbol{J}^-}{\partial q'^-} = \begin{bmatrix} 0 \\ 0 \\ -s \\ 0 \\ 0 \\ r^- c \end{bmatrix}, \tag{A.16}$$

$$\frac{\partial \boldsymbol{G}}{\partial q^+} = \frac{\partial \boldsymbol{J}^r}{\partial q^+} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -r^+ \\ 0 \end{bmatrix}, \frac{\partial \boldsymbol{G}}{\partial q^-} = \frac{\partial \boldsymbol{J}^r}{\partial q^-} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -r^- \\ 0 \\ 0 \end{bmatrix}, \tag{A.17}$$

$$\frac{\partial \boldsymbol{G}}{\partial q'^{+}} = \frac{\partial \boldsymbol{J}^{r}}{\partial q'^{+}} = \begin{bmatrix} -c \\ -s \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \frac{\partial \boldsymbol{G}}{\partial q'^{-}} = \frac{\partial \boldsymbol{J}^{r}}{\partial q'^{-}} = \begin{bmatrix} s \\ -c \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{A.18}$$

$$\frac{\partial \boldsymbol{J}'^{+}}{\partial q^{r}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \frac{\partial \boldsymbol{J}'^{+}}{\partial q^{+}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -r^{+}c \end{bmatrix}, \frac{\partial \boldsymbol{J}'^{+}}{\partial q^{-}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^{-}s \end{bmatrix}, \frac{\partial \boldsymbol{J}'^{+}}{\partial q'^{+}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -r'^{+} \end{bmatrix}, \tag{A.19}$$

$$\frac{\partial \boldsymbol{J}'^{-}}{\partial q^{r}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \frac{\partial \boldsymbol{J}'^{-}}{\partial q^{+}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^{+}s \end{bmatrix}, \frac{\partial \boldsymbol{J}'^{-}}{\partial q^{-}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^{-}c \end{bmatrix}, \frac{\partial \boldsymbol{J}'^{-}}{\partial q'^{-}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -r'^{-} \end{bmatrix}. \tag{A.20}$$

Appendix B

Derivatives for the 2D curve-curve contact

This Appendix calculates the twist and wrench Jacobians for the virtual contact manipulator of the *curve-curve* contact, as well as their derivatives with respect to their joint angles, in all three representation types discussed in the Chapter 3. In the general 2D contour following task, see FIG. 1.12 and Chapter 6, a smooth curve moves over another smooth curve. This is *cylinder-cylinder* limit case of the 3D *surface-surface* contact of Appendix A, with $\alpha = \pi$ and $r^- = r'^- = 0$. Only two degrees-of-freedom remain: rotations about the centres of curvature of both curves. Again, the end effector frame $\{ee\}$ and the base frame $\{bs\}$ are chosen to coincide with the principal frame $\{pri\}$. The following paragraphs explain how to interpret the derivatives. A completely similar interpretation exists for the *surface-surface* contact of Appendix A, but for the 2D case it is easier to draw the figures. The two degrees-of-freedom twist Jacobian is:

$$ {}_{bs}\boldsymbol{J}^{bs} = {}_{ee}\boldsymbol{J}^{ee} = {}_{bs}\boldsymbol{J}^{ee} = \left[\boldsymbol{J}'^{+}\ \boldsymbol{J}^{+}\right] = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ r'^{+} & -r^{+} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{B.1} $$

B.1 Hybrid representation

The hybrid representation takes the origin of the moving frame, $\{ee\}$ in the FIGS. B.1-B.3, as reference point for the twists and wrenches represented by the Jacobian columns, and uses the fixed frame $\{bs\}$ to express their components.

Slip Jacobian First, the influence of q^+ and q'^+ on $\boldsymbol{J}^+$ is investigated, FIG. B.1. Equation (A.15) gives:

$$ \frac{\partial \boldsymbol{J}^{+}}{\partial q^{+}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^{+} \end{bmatrix}, \frac{\partial \boldsymbol{J}^{+}}{\partial q'^{+}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^{+} \end{bmatrix}. \tag{B.2} $$

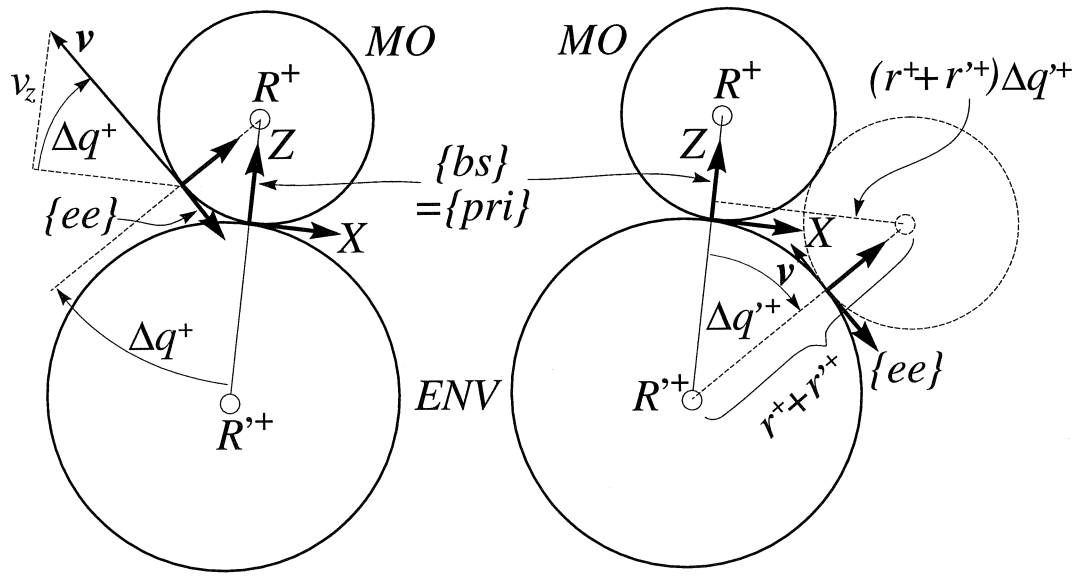


FIGURE B.1 : *Curve-curve contact: derivatives of the slip Jacobian.* Left: influence of a slip motion; right: influence of a slide motion.

Hence, $\boldsymbol{J}^+$ depends on Δq^+ as follows:

$$\boldsymbol{J}^+(\Delta q^+) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -r^+ \\ 0 \\ r^+ \Delta q^+ \end{bmatrix}. \tag{B.3}$$

This corresponds with the left-hand side of FIG. B.1: rotating the *MO* over a slip angle Δq^+ changes the direction of the tangential velocity $\boldsymbol{v}$, generated by a unit angular velocity of the slip joint R^+, by $[0\ 0\ \Delta v_z]^T$, with $\Delta v_z \approx r^+ \Delta q^+$.

$\boldsymbol{J}^+$ depends on $\Delta q'^+$ as follows:

$$\boldsymbol{J}^+\left(\Delta q'^+\right) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -r^+ \\ 0 \\ r^+ \Delta q'^+ \end{bmatrix}. \tag{B.4}$$

This corresponds with the right-hand side of FIG. B.1: rotating the *MO* over a slide angle $\Delta q'^+$ changes the direction of the tangential velocity $\boldsymbol{v}$, generated by a unit angular velocity of the slip joint R^+, by $[0\ 0\ \Delta v_z]^T$, with $\Delta v_z \approx r^+ \Delta q'^+$.

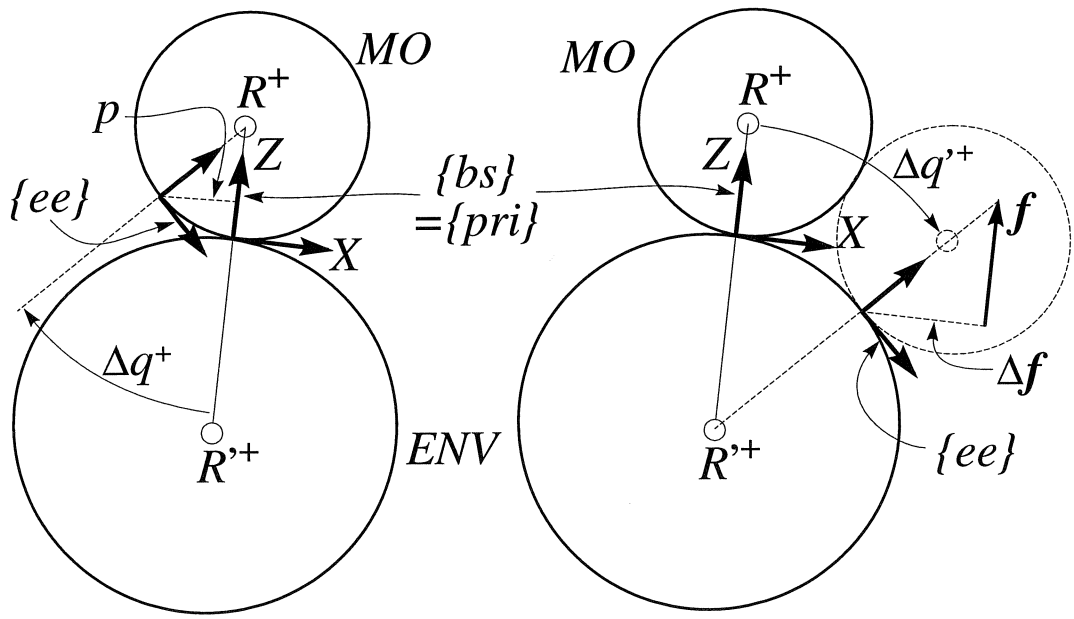


FIGURE B.2 : *Curve-curve contact: derivatives of the wrench Jacobian.* Left: influence of a slip motion; right: influence of a slide motion.

Wrench Jacobian The influence of q^+ and q'^+ on $\boldsymbol{G}$, FIG. B.2, is given by Eqs. (A.17) and (A.18):

$$\frac{\partial \boldsymbol{G}}{\partial q^+} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -r^+ \\ 0 \end{bmatrix}, \quad \frac{\partial \boldsymbol{G}}{\partial q'^+} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{B.5}$$

Hence, $\boldsymbol{G}$ depends on Δq^+ as follows:

$$\boldsymbol{G}(\Delta q^+) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -r^+ \Delta q^+ \\ 0 \end{bmatrix}. \tag{B.6}$$

This corresponds with the left-hand side of FIG. B.2: the ideal reaction force is not changed in space, i.e., it still lies along the Z axis of $\{bs\}$, but as seen from $\{ee\}$ this reaction force causes a moment component about the Y axis of $\{ee\}$, proportional to the lever arm p, with $p = r^+ \sin(\Delta q^+) \approx r^+ \Delta q^+$. The components of the reaction force three-vector and the moment three-vector are taken with respect to $\{bs\}$.

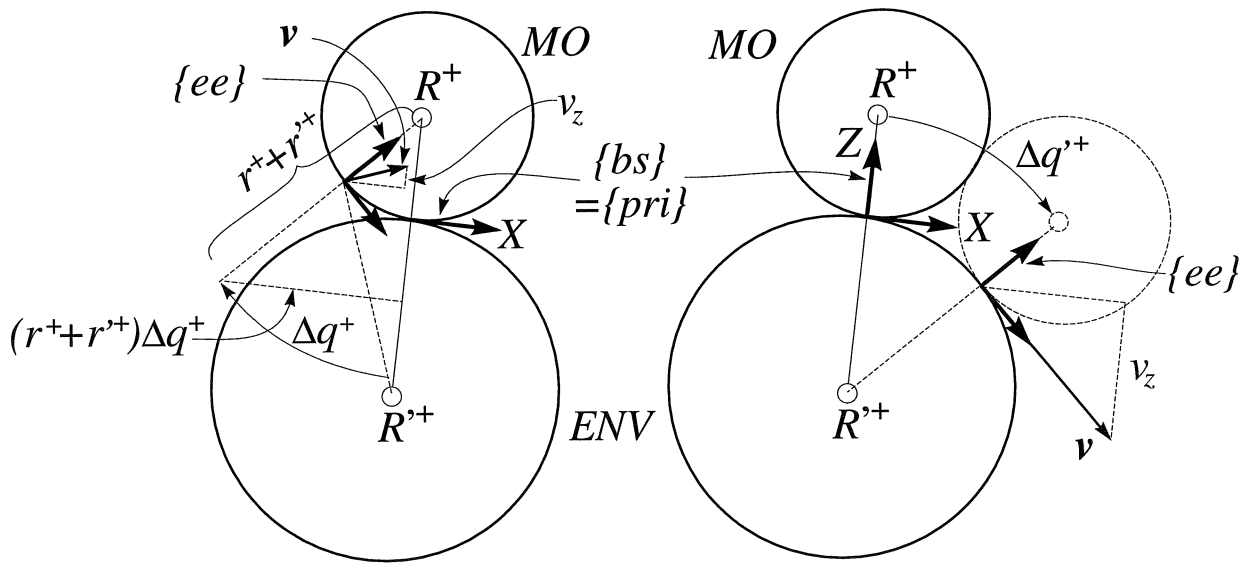


FIGURE B.3 : *Curve-curve contact: derivatives of the slide Jacobian.* Left: influence of a slip motion; right: influence of a slide motion.

$\boldsymbol{G}$ depends on $\Delta q'^{+}$ as follows:

$$\boldsymbol{G}\left(\Delta q'^{+}\right) = \begin{bmatrix} \Delta q'^{+} \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{B.7}$$

This corresponds with the right-hand side of FIG. B.2: the ideal reaction force is now effectively changed in space, and as seen from $\{ee\}$ the moment three-vector vanishes, and the force three-vector lies along the Z axis of $\{ee\}$. However, projecting this vector on $\{bs\}$ gives an X component $\Delta \boldsymbol{f}$ approximately proportional to $\Delta q'^{+}$

Slide Jacobian The influence of q^{+} and q'^{+} on $\boldsymbol{J}'^{+}$ is given by Eq. (A.19):

$$\frac{\partial \boldsymbol{J}'^{+}}{\partial q^{+}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^{+} \end{bmatrix}, \frac{\partial \boldsymbol{J}'^{+}}{\partial q'^{+}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -r'^{+} \end{bmatrix}. \tag{B.8}$$

Hence, $\boldsymbol{J}'^{+}$ depends on Δq^{+} as follows:

$$\boldsymbol{J}'^{+}\left(\Delta q^{+}\right) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ r'^{+} \\ 0 \\ r^{+}\Delta q^{+} \end{bmatrix}. \tag{B.9}$$

This corresponds with the left-hand side of FIG. B.3: rotating the MO over a slip angle Δq^{+} changes the direction of the velocity $\boldsymbol{v}$, generated by a unit angular velocity of the slide joint R'^{+}, by $[0\ 0\ \Delta v_z]^T$, with $\Delta v_z \approx r^{+}\Delta q^{+}$.

$\boldsymbol{J}'^{+}$ depends on $\Delta q'^{+}$ as follows:

$$\boldsymbol{J}'^{+}\left(\Delta q'^{+}\right) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ r^{+} \\ 0 \\ -r'^{+}\Delta q'^{+} \end{bmatrix}. \tag{B.10}$$

This corresponds with the right-hand side of FIG. B.3: rotating the MO over a slide angle $\Delta q'^{+}$ changes the direction of the tangential velocity $\boldsymbol{v}$, generated by a unit angular velocity of the slide joint R'^{+}, by $[0\ 0\ \Delta v_z]^T$, with $\Delta v_z \approx -r'^{+}\Delta q'^{+}$.

B.2 Inertial representation

The inertial representation takes the origin of the fixed frame, $\{bs\}$ in the FIGS. B.1-B.3, as reference point for the twists and wrenches represented by the Jacobian columns, and uses it to express their components too. The only two non-zero derivatives follow from Eqs. (A.7) and (A.8):

$$\frac{\partial \boldsymbol{G}}{\partial q'^{+}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -r'^{+} \\ 0 \end{bmatrix}, \quad \frac{\partial \boldsymbol{J}^{+}}{\partial q'^{+}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^{+} + r'^{+} \end{bmatrix}. \tag{B.11}$$

Hence:

$$\boldsymbol{G}\left(\Delta q'^{+}\right) = \begin{bmatrix} \Delta q'^{+} \\ 0 \\ 1 \\ 0 \\ -r'^{+}\Delta q'^{+} \\ 0 \end{bmatrix}, \quad \boldsymbol{J}^{+}\left(\Delta q'^{+}\right) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -r^{+} \\ 0 \\ \Delta q'^{+}\left(r^{+} + r'^{+}\right) \end{bmatrix}. \tag{B.12}$$

The first equality is compatible with the right-hand side of FIG. B.2: the ideal unit reaction force lies along the Z axis of $\{ee\}$, which gives a small X component of the force three-vector in the $\{bs\}$ frame, and a moment three-vector about the Y axis proportional to the lever arm between $\{bs\}$ and $\{ee\}$. This lever arm is $r'^{+}\sin(\Delta q'^{+}) \approx r'^{+}\Delta q'^{+}$.

The dependence of $\boldsymbol{J}^{+}$ on $\Delta q'^{+}$ is clarified by the right-hand side of FIG. B.1: the slip joint R^{+} is moved over an angle $\Delta q'^{+}$ about the slide joint R'^{+}, and hence the lever arm between this joint and the origin of the $\{bs\}$ frame is $(r^{+}+r'^{+})\sin(\Delta q'^{+}) \approx (r^{+}+r'^{+})\,\Delta q'^{+}$.

The other derivatives are zero. For example, $\boldsymbol{J}^{+}$ does not change with q^{+}, since a slip motion does not change the position of the slip joint R^{+} with respect to the base frame $\{bs\}$. Similarly for the influence of q'^{+} on $\boldsymbol{J}'^{+}$.

B.3 Body-fixed representation

The body-fixed representation takes the origin of the moving frame, $\{ee\}$ in the FIGS. B.1-B.3, as reference point for the twists and wrenches represented by the Jacobian columns, and uses it to express their components too. Only two derivatives are not identically zero, Eqs. (A.12) and (A.13):

$$\frac{\partial \boldsymbol{J}'^{+}}{\partial q^{+}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^{+}+r'^{+} \end{bmatrix}, \quad \frac{\partial \boldsymbol{G}}{\partial q^{+}} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -r^{+} \\ 0 \end{bmatrix}. \tag{B.13}$$

Hence:

$$\boldsymbol{J}'^{+}(\Delta q^{+}) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ r'^{+} \\ 0 \\ \Delta q^{+}(r^{+}+r'^{+}) \end{bmatrix}. \tag{B.14}$$

The left-hand side of FIG. B.3 explains this dependence of $\boldsymbol{J}'^{+}$ on Δq^{+}: the slide joint R'^{+} is moved over an angle $-\Delta q'^{+}$ about the slip joint R^{+}, and hence the lever arm $\boldsymbol{p}$ between this joint and the origin of the $\{ee\}$ frame is:

$$\boldsymbol{p} = \boldsymbol{R}(\Delta q^{+}) \begin{bmatrix} 0 \\ 0 \\ r^{+}+r'^{+} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ r^{+} \end{bmatrix} = \begin{bmatrix} s(r^{+}+r'^{+}) \\ 0 \\ -c(r^{+}+r'^{+}) - r^{+} \end{bmatrix}. \tag{B.15}$$

Hence, the translational velocity three-vector $\boldsymbol{v}$, FIG. B.3, is calculated as:

$$\boldsymbol{v} = \boldsymbol{\omega} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} c(r^{+}+r'^{+}) + r^{+} \\ 0 \\ s(r^{+}+r'^{+}) \end{bmatrix} \approx \begin{bmatrix} r'^{+} \\ 0 \\ \Delta q^{+}(r^{+}+r'^{+}) \end{bmatrix}. \tag{B.16}$$

The wrench Jacobian is influenced by q^+ as follows:

$$\boldsymbol{G}\left(\Delta q^+\right) = \begin{bmatrix} -\Delta q^+ \\ 0 \\ 1 \\ 0 \\ -\Delta q^+ r^+ \\ 0 \end{bmatrix}. \tag{B.17}$$

The interpretation follows straightforwardly from FIG. B.2: as seen from $\{ee\}$, the ideal reaction force gets an X component proportional to $\sin\left(-\Delta q^+\right) \approx -\Delta q^+$, and it causes a moment three-vector about the Y axis of $\{ee\}$ due to the lever arm p.

The other derivatives are zero. For example, $\boldsymbol{J}^+$ does not change with q^+ as seen from $\{ee\}$. Similarly for $\partial \boldsymbol{J}'^+/\partial q'^+$ and $\partial \boldsymbol{G}/\partial q'^+$.

Appendix C

Link with differential geometry

In the kinematic motion constraint models developed in Chapter 4, the principal frames $\{pri\}$ and $\{pri'\}$, on the *MO* and the *ENV* respectively, move due to the joint velocities of the virtual contact manipulator. This nominal motion of the principal frames is also described in the literature on differential geometry. For example, O'Neill (1966) gives the translational velocity of the principal frame's unit vectors $\boldsymbol{e}^X, \boldsymbol{e}^Y$ and $\boldsymbol{e}^Z$, expressed in the principal frame itself, when moving the origin of the frame with a unit velocity $\boldsymbol{v} = \cos(\theta)\boldsymbol{e}^X + \sin(\theta)\boldsymbol{e}^Y$. Using the nomenclature of this thesis, and the symbol "$\nabla_{\boldsymbol{v}}\boldsymbol{e}^i$" to denote the directional derivative of $\boldsymbol{e}^i$ in the direction $\boldsymbol{v}$, these *connection equations* are as follows:[1]

$$\begin{bmatrix} \nabla_{\boldsymbol{v}}\boldsymbol{e}^X \\ \nabla_{\boldsymbol{v}}\boldsymbol{e}^Y \\ \nabla_{\boldsymbol{v}}\boldsymbol{e}^Z \end{bmatrix} = \begin{bmatrix} 0 & a & \frac{\cos(\theta)}{r^+} \\ -a & 0 & \frac{\sin(\theta)}{r^-} \\ -\frac{\cos(\theta)}{r^+} & -\frac{\sin(\theta)}{r^-} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{e}^X \\ \boldsymbol{e}^Y \\ \boldsymbol{e}^Z \end{bmatrix}. \tag{C.1}$$

The parameter a depends on the *geodesic curvature* of the principal curves:

$$a = \cos(\theta)\kappa^+ + \sin(\theta)\kappa^-. \tag{C.2}$$

The geodesic curvature κ^+ (κ^-) is the angular velocity about $\boldsymbol{e}^Z$ of the principal frame if it moves with a unit velocity in the $\boldsymbol{e}^X$ ($\boldsymbol{e}^Y$) direction, *and* if the frame is kept aligned with the principal frames at the points it travels through.

The *slip* or *slide* parts of the virtual contact manipulator for smooth *MO* and *ENV* also describe how the principal frame can move over the surface. For example, the twist t of the principal frame $\{pri\}$ due to slip joint velocities $\dot{q}^+$ and $\dot{q}^-$ is given by:

$$\mathsf{t} = \dot{q}^+\boldsymbol{J}^+ + \dot{q}^-\boldsymbol{J}^- = \dot{q}^+ \begin{bmatrix} 0 \\ 1 \\ 0 \\ -r^+ \\ 0 \\ 0 \end{bmatrix} + \dot{q}^- \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ r^- \\ 0 \end{bmatrix}.$$

[1] Many thanks to S. Demey for the transformation of the very abstract description in O'Neill (1966) to the more understandable format of Eq. (C.1)!

In order to get the same translational velocity $\boldsymbol{v}$ as in Eq. (C.1), the joint velocities $\dot{q}^+$ and $\dot{q}^-$ must be equal to, respectively, $-\cos(\theta)/r^+$ and $\sin(\theta)/r^-$. Hence:

$$\mathsf{t} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \sin(\theta)/r^- \\ -\cos(\theta)/r^+ \\ 0 \\ \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix}. \tag{C.3}$$

$\boldsymbol{\nabla_v e}^X$ is defined to be the translational velocity of the endpoint of $\boldsymbol{e}^X$ minus the translational velocity of the origin of $\boldsymbol{e}^X$, expressed with respect to $\boldsymbol{e}^X, \boldsymbol{e}^Y, \boldsymbol{e}^Z$. The origin's motion corresponds to the translational velocity component of t. The endpoint's velocity is found from Eq. (3.5), since this relationship gives the velocity of another reference point on the same rigid body, which in this case is the endpoint of the unit vector $\boldsymbol{e}^X$. Hence:

$$\begin{aligned} \boldsymbol{\nabla_v e}^X &= (-\boldsymbol{e}^X \times \boldsymbol{\omega} + \boldsymbol{v}) - \boldsymbol{v} \\ &= -\boldsymbol{e}^X \times \boldsymbol{\omega} \\ &= \begin{bmatrix} 0 \\ 0 \\ \frac{\cos(\theta)}{r^+} \end{bmatrix}. \end{aligned}$$

Similar reasoning applies to the other components $\boldsymbol{\nabla_v e}^i$: $\boldsymbol{\nabla_v e}^Y = -\boldsymbol{e}^Y \times \boldsymbol{\omega}$, $\boldsymbol{\nabla_v e}^Z = -\boldsymbol{e}^Z \times \boldsymbol{\omega}$. This results in:

$$\begin{bmatrix} \boldsymbol{\nabla_v e}^X \\ \boldsymbol{\nabla_v e}^Y \\ \boldsymbol{\nabla_v e}^Z \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{\cos(\theta)}{r^+} \\ 0 & 0 & \frac{\sin(\theta)}{r^-} \\ -\frac{\cos(\theta)}{r^+} & -\frac{\sin(\theta)}{r^-} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{e}^X \\ \boldsymbol{e}^Y \\ \boldsymbol{e}^Z \end{bmatrix}. \tag{C.4}$$

This corresponds to Eq. (C.1), except for the components due to the geodesic curvature. The reason for this discrepancy is obvious: the second order kinematic model cannot predict how the directions of minimum and maximum curvature will change over the surface since this geodesic curvature is a geometric third order concept.

The discussion in the above paragraphs can be repeated if the MO is not a smooth surface, but a smooth curve. The role of the principal frame is now taken by the so-called *Frenet* frame $\{\boldsymbol{t}, \boldsymbol{b}, \boldsymbol{n}\}$, FIG. 4.4: $\boldsymbol{t}$ is the unit vector along the tangent to the curve; $\boldsymbol{n}$ is the normal vector, i.e., the vector normal to $\boldsymbol{t}$ and pointing to the center of the local osculating circle; $\boldsymbol{b}$ is the bi-normal vector, completing the right-handed orthogonal frame. According to the conventions of Section 4.3, $\boldsymbol{t} = \boldsymbol{e}^X, \boldsymbol{b} = \boldsymbol{e}^Y$ and $\boldsymbol{n} = \boldsymbol{e}^Z$. The Frenet frame can translate along the curve —i.e., a velocity along $\boldsymbol{e}^X$— and rotate about the tangent

—i.e., an angular velocity about $\boldsymbol{e}^X$. These two degrees-of-freedom are contained in the *slip* part of the twist Jacobian:

$$\boldsymbol{J}^{slip} = [\boldsymbol{J}^+ \ \boldsymbol{J}^-] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ -r^+ & 0 \\ 0 & r^- \to 0 \\ 0 & 0 \end{bmatrix}.$$

This slip Jacobian can also be represented by a slightly different basis:

$$\boldsymbol{J}^{slip} = \begin{bmatrix} 0 & 1 \\ -\kappa & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

$\kappa = 1/r^+$ is the curvature of the curve. A unit velocity along the tangent ($\boldsymbol{v} = \boldsymbol{e}^X$) is obtained by driving the first slip joint with a unit joint velocity. The resulting twist of the MO is then:

$$\mathsf{t} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = \boldsymbol{J}^+.$$

As for the principal frame, $\boldsymbol{\nabla}_{\boldsymbol{v}} \boldsymbol{e}^i, i = X, Y, Z$ equals $-\boldsymbol{e}^i \times \boldsymbol{\omega}$. The connection equations for the Frenet frame are given in the differential geometry literature, O'Neill (1966):

$$\begin{bmatrix} \boldsymbol{\nabla}_{\boldsymbol{v}} \boldsymbol{e}^X \\ \boldsymbol{\nabla}_{\boldsymbol{v}} \boldsymbol{e}^Y \\ \boldsymbol{\nabla}_{\boldsymbol{v}} \boldsymbol{e}^Z \end{bmatrix} = \begin{bmatrix} 0 & 0 & \kappa \\ 0 & 0 & -\tau \\ -\kappa & \tau & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{e}^X \\ \boldsymbol{e}^Y \\ \boldsymbol{e}^Z \end{bmatrix}. \tag{C.5}$$

τ is the curve's *torsion*, i.e., the angular velocity with which the curve rotates about the tangent when it translates along the tangent with a unit velocity. Torsion is a third order concept which was not included in the second order Jacobians of Chapter 4. From Eq. (C.5) it turns out that the torsion is very simply added to these second order models. Hence, the slip Jacobian becomes:

$$\boldsymbol{J}^{slip} = \begin{bmatrix} -\tau & 1 \\ -\kappa & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{C.6}$$

The previous paragraphs looked only at the translational velocities of reference points on the principal frame. *Euler's formula*, O'Neill (1966), gives the *normal curvature* κ of the

surface in the motion direction $\boldsymbol{v}$ as a function of the principal curvatures $\kappa^+ = 1/r^+$ and $\kappa^- = 1/r^-$:

$$\kappa(\boldsymbol{v}) = \cos^2(\theta)\kappa^+ + \sin^2(\theta)\kappa^- = \frac{\cos^2(\theta)}{r^+} + \frac{\sin^2(\theta)}{r^-}. \tag{C.7}$$

This equation can also be derived from the kinematic model, see Section 4.2.4.

Appendix D

Peg-on-hole

This Appendix gives mathematical details for the *peg-on-hole* contact situation, FIG. 4.6, which have not been included in the main text for reasons of readability.

Before a nominal motion can be specified based on the kinematic model with the three virtual contact manipulators of FIG. 4.7, this kinematic model must be *assembled* first, i.e., all fifteen joints must be placed in a position consistent with the current contact configuration. When the peg is tilted over an angle α with respect to the hole, FIG. 4.8, the displacement of the frame $\{ee\}$ with respect to the frame $\{bs\}$ is given by a homogeneous transformation matrix ${}^{ee}_{bs}\boldsymbol{T}$ of the following form:

$$ {}^{ee}_{bs}\boldsymbol{T} = \begin{bmatrix} c_\alpha & 0 & -s_\alpha & p_x \\ 0 & 1 & 0 & 0 \\ s_\alpha & 0 & c_\alpha & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{D.1} $$

with $c_\alpha = \cos(\alpha), s_\alpha = \sin(\alpha)$, and p_x and p_z the (yet unknown) X and Z components of the origin of $\{ee\}$ as seen from $\{bs\}$. The screw transformation matrix from $\{ee\}$ to $\{bs\}$ is:

$$ {}^{ee}_{bs}\boldsymbol{S} = \begin{bmatrix} c_\alpha & 0 & -s_\alpha & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ s_\alpha & 0 & c_\alpha & 0 & 0 & 0 \\ 0 & -p_z & 0 & c_\alpha & 0 & -s_\alpha \\ p_z c_\alpha - p_x s_\alpha & 0 & -p_z s_\alpha - p_x c_\alpha & 0 & 1 & 0 \\ 0 & p_x & 0 & s_\alpha & 0 & c_\alpha \end{bmatrix}. \tag{D.2} $$

The joints R^-, R^r and R'^- of both *curve-curve* virtual contact manipulators are situated at angles β or $-\beta$ from the reference points on the rims of peg and hole, as indicated in FIG. 4.8. The parameters α, β, h, p_x and p_z suffice to calculate all three twist Jacobians. First of all, the following subsections derive how β, h, p_x and p_z depend on α. This derivation is often called *position closure* in the kinematics literature. Then, the Jacobians are calculated, and the *velocity closure* equations, Eqs. (4.18), are constructed.

D.1 Position closure for the surface-curve contact

The positions of the joints for the *surface-curve* contact are as follows:

1. R'^+ lies at the origin of $\{bs\}$.

2. R'^{-} and R^{r} lie at $[-1\ 0\ 0]^T$. (Remember that peg and hole have unit radius.)

3. R^{+} and P^{-} lie at $[0\ 0\ h]^T$ with respect to $\{ee\}$.

4. R^{r} lies at $[-1\ 0\ h]^T$ with respect to $\{ee\}$.

Closing the links starting from $\{bs\}$ and $\{ee\}$ at the R^{r} joint, one gets the following position closure condition:

$$\begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = {}^{ee}_{bs}\boldsymbol{T} \begin{bmatrix} -1 \\ 0 \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} -c_\alpha - s_\alpha h + p_x \\ 0 \\ -s_\alpha + c_\alpha h + p_z \\ 1 \end{bmatrix},$$

$$\text{or} \quad \begin{cases} h = s_\alpha(1 + p_x) - c_\alpha p_z, \\ 1 = c_\alpha(1 + p_x) + s_\alpha p_z. \end{cases} \tag{D.3}$$

D.2 Position closure for the curve-curve contact

Similarly for one of the *curve-curve* virtual contact manipulators:

1. R'^{+} lies at the origin of $\{bs\}$.

2. R'^{-}, R^{r} and R^{-} lie at $[-c_\beta\ -s_\beta\ 0]^T$, with $c_\beta = \cos(\beta)$ and $s_\beta = \sin(\beta)$.

3. R^{+} lies at the origin of $\{ee\}$.

4. R'^{-}, R^{r} and R^{-} lie at $[c_\beta\ -s_\beta\ 0]^T$ with respect to $\{ee\}$.

A second set of position closure conditions result:

$$\begin{bmatrix} -c_\beta \\ -s_\beta \\ 0 \\ 1 \end{bmatrix} = {}^{ee}_{bs}\boldsymbol{T} \begin{bmatrix} c_\beta \\ -s_\beta \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} c_\alpha c_\beta + p_x \\ -s_\beta \\ s_\alpha c_\beta + p_z \\ 1 \end{bmatrix},$$

$$\text{or} \quad \begin{cases} p_x = -c_\beta(1 + c_\alpha), \\ p_z = -s_\alpha c_\beta. \end{cases} \tag{D.4}$$

The other *curve-curve* virtual contact manipulator gives the same position closure conditions.

D.3 Total position closure

Equations (D.3) and (D.4) together yield:

$$\begin{array}{rcl} p_x &=& 1 - c_\alpha, \\ p_z &=& -s_\alpha c_\beta, \\ h &=& \frac{2s_\alpha}{1+c_\alpha}, \\ c_\beta &=& \frac{c_\alpha - 1}{c_\alpha + 1}. \end{array} \tag{D.5}$$

With these results, the screw transformation, Eq. (D.2), reduces to:

$${}^{ee}_{bs}\boldsymbol{S} = \begin{bmatrix} c_\alpha & 0 & -s_\alpha & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ s_\alpha & 0 & c_\alpha & 0 & 0 & 0 \\ 0 & -p_z & 0 & c_\alpha & 0 & -s_\alpha \\ -p_z & 0 & -p_x & 0 & 1 & 0 \\ 0 & p_x & 0 & s_\alpha & 0 & c_\alpha \end{bmatrix}. \tag{D.6}$$

D.4 Jacobian for the surface-curve contact

The previous paragraphs calculated the positions of all joints; in order to calculate the twist Jacobians, one also needs to know the joints' orientations. Again, this information follows straightforwardly from FIG. 4.8. For the *surface-curve* contact:

1. R'^{+} rotates about the Z axis of $\{bs\}$.
2. The orientation of R'^{-} is $[0\ -1\ 0]^T$, with respect to $\{bs\}$.
3. The orientation of R^r is $[c_\alpha\ 0\ s_\alpha]^T$, with respect to $\{bs\}$.
4. The orientation of R^+ and P^- is $[0\ 0\ 1]^T$ with respect to $\{ee\}$.

Equation (3.12) gives the Jacobians in the inertial representation. Transformation of these Jacobians to other representations is explained in Chapter 3. For the *surface-curve* contact this gives:

$${}_{ee}\boldsymbol{J}^{+} = {}_{bs}\boldsymbol{J}'^{+} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}_{ee}\boldsymbol{J}^{-} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$${}_{bs}\boldsymbol{J}'^{-} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad {}_{bs}\boldsymbol{J}^{r} = \begin{bmatrix} c_\alpha \\ 0 \\ s_\alpha \\ 0 \\ s_\alpha \\ 0 \end{bmatrix}.$$

Transforming ${}_{ee}\boldsymbol{J}^{+}$ and ${}_{ee}\boldsymbol{J}^{-}$ with the screw transformation ${}^{ee}_{bs}\boldsymbol{S}$, Eq. (D.6), into ${}_{bs}\boldsymbol{J}^{+}$ and ${}_{bs}\boldsymbol{J}^{-}$ yields:

$$
\begin{aligned}
{}_{bs}\boldsymbol{J} &= \left[\boldsymbol{J}'^{+}\ \boldsymbol{J}'^{-}\ \boldsymbol{J}^{r}\ \boldsymbol{J}^{-}\ \boldsymbol{J}^{+}\right] \\
&= \begin{bmatrix}
0 & 0 & c_\alpha & 0 & -s_\alpha \\
0 & -1 & 0 & 0 & 0 \\
1 & 0 & s_\alpha & 0 & c_\alpha \\
0 & 0 & 0 & -s_\alpha & 0 \\
0 & 0 & s_\alpha & 0 & -p_z s_\alpha - p_x c_\alpha \\
0 & 1 & 0 & c_\alpha & 0
\end{bmatrix}.
\end{aligned} \tag{D.7}
$$

D.5 Jacobians for the curve-curve-contacts

Similarly for one of the *curve-curve* contacts:

1. R'^{+} rotates about the Z axis of $\{bs\}$.
2. The orientation of R'^{-} is $[s_\beta\ -c_\beta\ 0]^T$, with respect to $\{bs\}$.
3. The orientation of R^{+} is $[0\ 0\ 1]^T$ with respect to $\{ee\}$.
4. The orientation of R^{-} is $[-s_\beta\ -c_\beta\ 0]^T$ with respect to $\{ee\}$.
5. R^{r} lies along the cross product of the joint axes of R'^{-} and R^{-}.

Hence:

$$
{}_{ee}\boldsymbol{J}^{+} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},\ {}_{ee}\boldsymbol{J}^{-} = \begin{bmatrix} -s_\beta \\ -c_\beta \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix},
$$

$$
{}_{bs}\boldsymbol{J}'^{+} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},\ {}_{bs}\boldsymbol{J}'^{-} = \begin{bmatrix} s_\beta \\ -c_\beta \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.
$$

$\boldsymbol{J}^{r}$ is calculated as follows: 1) its angular velocity three-vector is the cross product of the angular velocity three-vectors of $\boldsymbol{J}^{-}$ and $\boldsymbol{J}'^{-}$; 2) its translational velocity three-vector is the cross product of its angular velocity three-vector with the vector from the origin of R^{r} to the origin of $\{bs\}$.

With Eqs. (D.5) and (D.6), and some rewriting, one finds:

$$
\begin{aligned}
{}_{bs}\boldsymbol{J} &= \begin{bmatrix} \boldsymbol{J}'^{+} & \boldsymbol{J}'^{-} & \boldsymbol{J}^{r} & \boldsymbol{J}^{-} & \boldsymbol{J}^{+} \end{bmatrix} \\
&= \begin{bmatrix}
0 & s_\beta & s_\alpha c_\beta s_\beta & -c_\alpha s_\beta & -s_\alpha \\
0 & -c_\beta & s_\alpha s_\beta^2 & -c_\beta & 0 \\
1 & 0 & -s_\beta(c_\alpha - 1) & -s_\alpha s_\beta & c_\alpha \\
0 & 0 & s_\beta^2(c_\alpha - 1) & s_\alpha s_\beta^2 & 0 \\
0 & 0 & -c_\beta s_\beta(c_\alpha - 1) & -s_\alpha c_\beta s_\beta & -p_x \\
0 & 1 & 0 & c_\beta^2 - c_\alpha s_\beta^2 & 0
\end{bmatrix}.
\end{aligned} \tag{D.8}
$$

D.6 Velocity closure

The motion freedom of the peg, while maintaining the three contacts with the hole, is given by the intersection of the twist spaces spanned by the Jacobians of the three virtual contact manipulators. These Jacobians have been calculated in the previous paragraphs. The leading subscript "*bs*," indicating that the twists are expressed with respect to the base reference frame $\{bs\}$, is omitted further-on, since all quantities refer to this frame. The subscript is replaced by either "*sc*," "*cc*1" or "*cc*2," referring to the *surface-curve* contact, or both *curve-curve* contacts. This gives the velocity closure equations as described in Eqs. (4.18). The symmetry considerations of Section 4.4.3 simplify these equations considerably:

1. Only one row of the velocity closure equations must be taken into consideration:

$$\boldsymbol{J}_{sc}\,\dot{\boldsymbol{q}}_{sc} - \boldsymbol{J}_{cc1}\,\dot{\boldsymbol{q}}_{cc1} = 0. \tag{D.9}$$

   The second row is automatically satisfied with $\dot{\boldsymbol{q}}_{cc2}$ deduced from $\dot{\boldsymbol{q}}_{cc1}$ by the symmetry arguments.

2. Since $\dot{q}'^{+}_{sc} = \dot{q}^{+}_{sc} = \dot{q}^{r}_{sc} = 0$, $\dot{q}^{-}_{sc} = \dot{h}$ and $\dot{q}'^{-}_{sc} = \dot{\alpha}$, Eq. (D.9) further reduces to:

$$\dot{h}\boldsymbol{J}^{-}_{sc} - \dot{q}'^{+}_{cc1}\boldsymbol{J}'^{+}_{cc1} - \dot{q}'^{-}_{cc1}\boldsymbol{J}'^{-}_{cc1} - \dot{q}^{r}_{cc1}\boldsymbol{J}^{r}_{cc1} - \dot{q}^{-}_{cc1}\boldsymbol{J}^{-}_{cc1} - \dot{q}^{+}_{cc1}\boldsymbol{J}^{+}_{cc1} = -\dot{\alpha}\boldsymbol{J}'^{-}_{sc}, \tag{D.10}$$

$$
\begin{aligned}
\text{or}\quad \dot{\alpha}\begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} &= -\dot{h}\begin{bmatrix} 0 \\ 0 \\ 0 \\ -s_\alpha \\ 0 \\ c_\alpha \end{bmatrix} + \dot{q}'^{+}_{cc1}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \dot{q}'^{-}_{cc1}\begin{bmatrix} s_\beta \\ -c_\beta \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
&+\dot{q}^{r}_{cc1}\begin{bmatrix} s_\alpha c_\beta s_\beta \\ s_\alpha s_\beta^2 \\ s_\beta(1-c_\alpha) \\ -s_\beta^2(1-c_\alpha) \\ c_\beta s_\beta(1-c_\alpha) \\ 0 \end{bmatrix} + \dot{q}^{-}_{cc1}\begin{bmatrix} -c_\alpha s_\beta \\ -c_\beta \\ -s_\alpha s_\beta \\ s_\alpha s_\beta^2 \\ -s_\alpha c_\beta s_\beta \\ c_\beta^2 - c_\alpha s_\beta^2 \end{bmatrix} + \dot{q}^{+}_{cc1}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.
\end{aligned}
$$

This set of six equations in six unknowns can be solved analytically. Its solution reveals two more symmetries:

$$\dot{h} = (c_\beta - 1)\dot{\alpha}, \tag{D.11}$$

$$\dot{q}'^{+} = \dot{q}^{+} = \frac{s_\alpha s_\beta}{2c_\alpha}\dot{\alpha} \tag{D.12}$$

$$\dot{q}'^{-} = \dot{q}^{-} = -\frac{\dot{\alpha}}{2}, \tag{D.13}$$

$$\dot{q}^{r} = \frac{s_\alpha \dot{\alpha}}{2c_\alpha(c_\alpha - 1)}. \tag{D.14}$$

The velocity of the peg, resulting from applying a driving velocity $\dot{\alpha}$, is then found as:

$$\mathsf{t} = \dot{\alpha}\begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \dot{h}\begin{bmatrix} 0 \\ 0 \\ 0 \\ -s_\alpha \\ 0 \\ c_\alpha \end{bmatrix} = \dot{\alpha}\begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ s_\alpha(1 - c_\beta) \\ 0 \\ -c_\beta \end{bmatrix}. \tag{D.15}$$

This twist is expressed in the inertial representation, i.e., with full indexing it should be written ${}_{bs}\mathsf{t}^{bs}$, Chapter 3. Transforming it into the hybrid representation gives:

$${}_{bs}\mathsf{t}^{ee} = \boldsymbol{M}_{bs}^{ee}\ {}_{bs}\mathsf{t}^{bs} = \dot{\alpha}\begin{bmatrix} 0 \\ -1 \\ 0 \\ s_\alpha \\ 0 \\ -(2 + c_\alpha)c_\beta \end{bmatrix}. \tag{D.16}$$

It is easily checked that the translational velocity three-vector of this twist corresponds to the time derivative of the position vector from the origin of the hole reference frame $\{bs\}$ to the origin of the peg reference frame $\{ee\}$, i.e., $\boldsymbol{p}^{bs,ee} = [p_x\ 0\ p_z]^T$, Eq. (D.5).

Appendix E

Incompatible 3D seam following

This appendix describes the *incompatible 3D seam following* task, FIG. E.1: a tool makes two *vertex-surface* contacts with the "seam," which consists of two smooth and intersecting surfaces. The task is called "incompatible" because the tool has a different geometry than the seam. The goal of the task is to maintain the contacts and move along the seam in a specified way. The first Section defines the relevant geometric parameters of the task. The following Section describes a basis for the degrees-of-freedom that can be used for an unambiguous motion specification, independent of any choice of reference frame. This motion specification is discussed in the final Section.

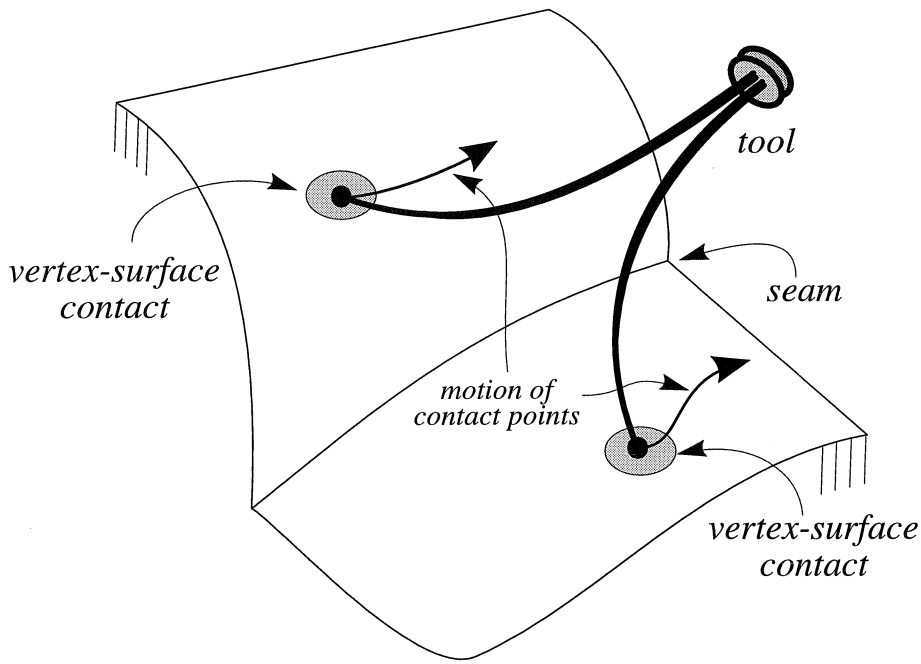


FIGURE E.1 : *Incompatible 3D seam following.*

E.1 Definitions of geometric parameters

The symbols {1} and {2} denote the two contact points, as well as their associated contact frames $X^1Y^1Z^1$ and $X^2Y^2Z^2$, FIGS. E.2 and E.3. The *seam axis* is the intersection of the

two tangent planes at $\{1\}$ and $\{2\}$. A *normal plane* is each plane through a contact normal and parallel to the seam axis. A *transverse plane* is any plane perpendicular to the seam axis. The *seam angle* σ is the (free space) angle between the tangent planes. The contact points lie at distances d^1 and d^2, respectively, from the seam axis.

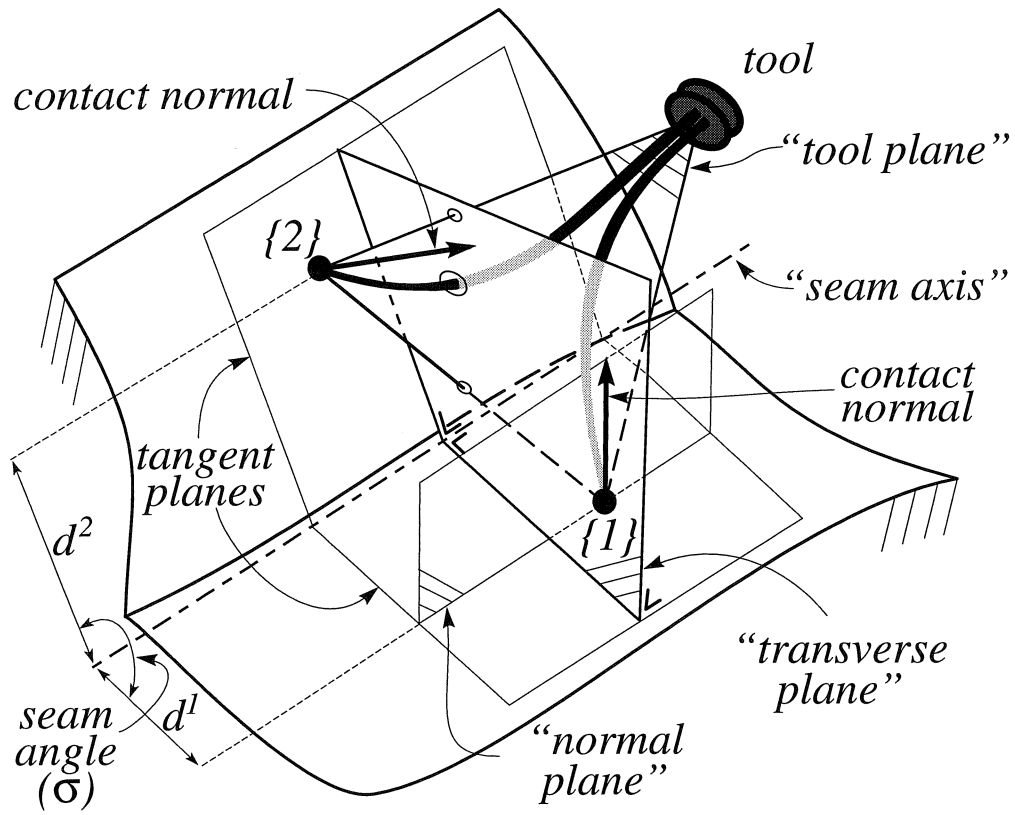


FIGURE E.2 : *3D seam following: geometric definitions.*

All these parameters can be calculated if the tangent planes in both contacts are known. Assume that the unit normal vectors $\boldsymbol{e}^{Z^1}$ and $\boldsymbol{e}^{Z^2}$ are known, as well as the vector $\boldsymbol{p}$ linking the two contact points, with respect to the reference frame $\{1\}$ of FIG. E.3:

$$\boldsymbol{e}^{Z^1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{e}^{Z^2} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}, \quad \boldsymbol{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}. \tag{E.1}$$

Then, the seam angle σ, the direction of the seam axis $\boldsymbol{e}^{sa}$, the position vectors $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$ of the points on the seam axis closest to $\{1\}$ and $\{2\}$, as well as the distances d^1 and d^2 from the contact points to the seam axis, are calculated as follows:[1]

[1]This representation of the contact situation partially degenerates if both tangent planes are parallel.

1. The seam angle σ:

$$\sigma = \arcsin\left(\left(\boldsymbol{e}^{Z^1}\right)^T \boldsymbol{e}^{Z^2}\right) + \frac{\pi}{2} = \arcsin(e_z) + \frac{\pi}{2}.$$

2. The vector $\boldsymbol{e}^{sa}$ is parallel to the cross product of $\boldsymbol{e}^{Z^2}$ and $\boldsymbol{e}^{Z^1}$:

$$\boldsymbol{e}^{sa} = \frac{\boldsymbol{e}^{Z^2} \times \boldsymbol{e}^{Z^1}}{\mid \boldsymbol{e}^{Z^2} \times \boldsymbol{e}^{Z^1} \mid} = \frac{1}{\sqrt{e_x^2 + e_y^2}} \begin{bmatrix} e_y \\ -e_x \\ 0 \end{bmatrix}.$$

3. $\boldsymbol{p}_d^1$ is the intersection of the tangent planes at $\{1\}$ and $\{2\}$, and the transverse plane through $\{1\}$:

$$\begin{cases} 0x + 0y + z &= 0 \\ e_x x + e_y y + e_z z &= e_x p_x + e_y p_y + e_z p_z \\ e_y x - e_x y + 0z &= 0 \end{cases}$$

$$\Rightarrow \boldsymbol{p}_d^1 = \frac{e_x p_x + e_y p_y + e_z p_z}{e_x^2 + e_y^2} \begin{bmatrix} e_x \\ e_y \\ 0 \end{bmatrix}.$$

Similarly, $\boldsymbol{p}_d^2$ is the intersection of the tangent planes at $\{1\}$ and $\{2\}$, and the transverse plane through $\{2\}$:

$$\begin{cases} 0x + 0y + z &= 0 \\ e_x x + e_y y + e_z z &= e_x p_x + e_y p_y + e_z p_z \\ e_y x - e_x y + 0z &= e_y p_x - e_x p_y \end{cases}$$

$$\Rightarrow \boldsymbol{p}_d^2 = \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix} + \frac{e_z p_z}{e_x^2 + e_y^2} \begin{bmatrix} e_x \\ e_y \\ 0 \end{bmatrix}.$$

4. The distance d^1 is the length of $\boldsymbol{p}_d^1$:

$$d^1 = \frac{\mid e_x p_x + e_y p_y + e_z p_z \mid}{\sqrt{e_x^2 + e_y^2}},$$

The distance d^2 is the length of $\boldsymbol{p}_d^2 - \boldsymbol{p}$:

$$d^2 = \mid p_z \mid \sqrt{\frac{e_x^2 + e_y^2 + e_z^2}{e_x^2 + e_y^2}}.$$

The *tool plane* is defined as the plane through the two contact points and a third user-defined point $\boldsymbol{p}^t$ on the tool. The tool plane has two special reference positions: 1) it lies in a transverse plane, and 2) $d^1 = d^2$. The first situation is called the *transverse tool* position, the second one is called the *symmetric tool* position. Both positions are not mutually exclusive. The tool is in a singular position if it is *parallel* to the seam, i.e., $d^1 = d^2 = 0$.

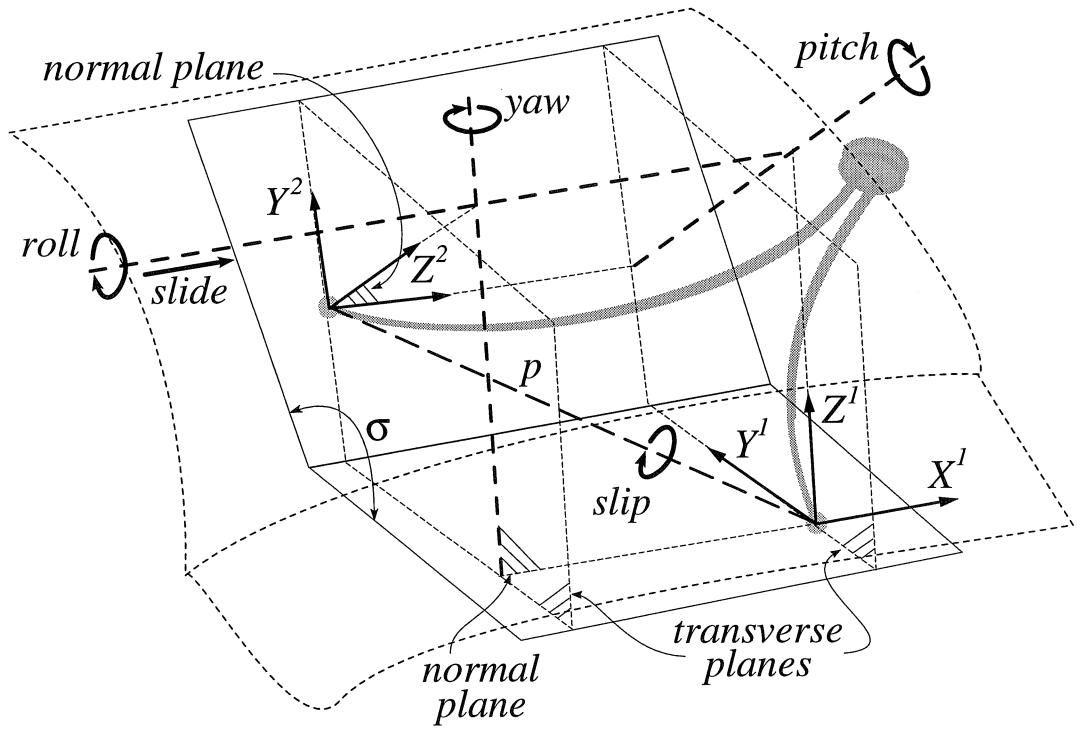


FIGURE E.3 : *3D seam following: motion degrees-of-freedom.* The four remaining degrees-of-freedom can be described by rotations about, and translations along, geometrically defined lines: *roll, pitch, yaw,* and *slide.*

E.2 Motion constraint model

The contact frames at each of the two contact points have their Z axis along the contact normal. The X and Y axes are not uniquely determined geometrically; their direction can be freely chosen in the tangent plane. A contact frame is called *parallel* if its X axis is parallel to the seam axis, FIG. E.3. Each of the two contacts reduces the *MO*'s motion freedom by one. With respect to parallel reference frame in $\{1\}$, the Jacobians of the two contacts in the inertial representation (Chapter 3) are:

$$\boldsymbol{J}^1 = \begin{array}{c} \begin{matrix} \textit{slide} & & \textit{rot} & \textit{slip} & \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}, \tag{E.2}$$

$$\boldsymbol{J}^2 = \begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \multicolumn{2}{c}{slide} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} rot \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \multicolumn{2}{c}{slip} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -s_\sigma & 0 & -c_\sigma \\ 0 & 0 & -c_\sigma & 0 & s_\sigma \\ 1 & 0 & -p_y c_\sigma + p_z s_\sigma & 0 & p_y s_\sigma + p_z c_\sigma \\ 0 & -c_\sigma & p_x c_\sigma & p_z & -p_x s_\sigma \\ 0 & s_\sigma & -p_x s_\sigma & -p_y & -p_x c_\sigma \end{bmatrix}. \tag{E.3}$$

c_σ and s_σ are the cosine and sine of the seam angle σ. Subtracting simple multiples of the second column from the three last columns simplifies $\boldsymbol{J}^2$ to an equivalent basis:

$$\boldsymbol{J}^2 \cong \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -s_\sigma & 0 & -c_\sigma \\ 0 & 0 & -c_\sigma & 0 & s_\sigma \\ 1 & 0 & -p_y c_\sigma + p_z s_\sigma & 0 & p_y s_\sigma + p_z c_\sigma \\ 0 & -c_\sigma & 0 & p_z - p_y \frac{c_\sigma}{s_\sigma} & -\frac{p_x}{s_\sigma} \\ 0 & s_\sigma & 0 & 0 & 0 \end{bmatrix}. \tag{E.4}$$

Comparing Eqs.(E.2) and (E.4), it is clear that $\boldsymbol{J}^{tot}$, the Jacobian matrix of the total constraint, is $\boldsymbol{J}^2$ with the second column removed, since this column is the only one with a v_z component. Removing this column and some rewriting gives:

$$\begin{aligned} \boldsymbol{J}^{tot} &\cong \begin{bmatrix} \boldsymbol{J}_4^2 & \boldsymbol{J}_3^2 & \left(s_\sigma \boldsymbol{J}_5^2 - c_\sigma \boldsymbol{J}_3^2 - p_y \boldsymbol{J}_1^2\right) & \boldsymbol{J}_1^2 \end{bmatrix} \\ &= \begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} roll \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} pitch \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} yaw \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} slide \\ \\ \\ \\ \\ \\ \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -s_\sigma & 0 & 0 \\ 0 & -c_\sigma & 1 & 0 \\ 0 & -p_y c_\sigma + p_z s_\sigma & 0 & 1 \\ p_z - p_y \frac{c_\sigma}{s_\sigma} & 0 & -p_x & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{E.5}$$

These four remaining degrees-of-freedom represent rotations about, and translations along, geometrically defined lines, FIG. E.3:

1. *Roll* is rotation about the intersection of the normal planes through {1} and {2}.
2. *Pitch* is rotation about the intersection of the transverse plane in {1} and the normal plane in {2}.
3. *Yaw* is the rotation about the intersection of the normal plane in {1} and the transverse plane in {2}.
4. *Slide* is translation in the direction of the seam axis. In accordance with the previous definitions, it can be considered as the rotation about the intersection of the transverse planes in {1} and {2}; this intersection lies at infinity, hence the rotation becomes a translation.

Roll, pitch and yaw have (almost) their original maritime interpretation if one looks at the *MO* as a "ship" travelling along the seam. However, their axes do not intersect in one point! Another interesting degree-of-freedom is the rotation about the line between the two contact points. This motion leaves the contact points unchanged on the *ENV*, hence it is called *slip*. Slip is a linear combination of roll, pitch, yaw and slide:

$$\begin{aligned} \boldsymbol{J}^{slip} &= p_x\, \boldsymbol{J}^{roll} - \tfrac{p_y}{s_\sigma}\, \boldsymbol{J}^{pitch} \\ &+ \left(p_z - p_y \tfrac{c_\sigma}{s_\sigma}\right)\, \boldsymbol{J}^{yaw} - (p_z s_\sigma - p_y c_\sigma)\, \boldsymbol{J}^{slide} \\ &= \begin{bmatrix} p_x \\ p_y \\ p_z \\ 0 \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{E.6}$$

The *roll-pitch-yaw-slide* and *roll-pitch-yaw-slip* bases are completely and unambiguously determined by the geometry of the *ENV*, and hence independent of any choice of reference frames. Remark that these bases are time-varying, i.e., the relative positions of the axes change during the motion, due to the curvature of the contact surfaces.

E.3 Motion specification

The *MO*'s motion consists of two parts: 1) move the *MO* along the seam, and 2) change its orientation with respect to the local seam geometry. The former motion uses mainly *slide* velocity set-points, the latter mainly *roll, pitch, yaw* or *slip* velocity set-points.

Since the contact situation is a non-holonomic system, the desired *orientation* can *not* be defined by roll, pitch and yaw *angles* of the tool with respect to, for example, the *symmetric* and *transverse tool* position. *Angular velocity* set-points should be used instead. These velocities must be chosen in such a way that they bring the tool closer to a desired relative position with respect to the seam.

One possible position set-point specification approach is given here:

1. Roll is specified by giving a desired distance d_1 between the first contact point and the seam.

2. Slip is specified by giving a desired height p_z^t of the tool point p^t above the tangent plane at $\{1\}$.

3. Yaw is specified by giving a desired distance p_x^2 along the seam between $\{1\}$ and $\{2\}$.

4. Slide is specified by giving a desired travelled distance along the seam, or alternatively a desired velocity along the seam,

These position set-points can be time-varying. Any deviations between desired and measured positions must be transformed in velocity set-points in the four motion degrees-of-freedom, by means of a user-specified control law.